Graz University of Technology

Master's Thesis

Channel Equalization for 13.56 MHz RFID Systems

conducted at the
Signal Processing and Speech Communications Laboratory
Graz University of Technology, Austria

in co-operation with
NXP Semiconductors
Gratkorn, Austria

by
Tim Raspel, 01031306

Supervisors:
Assoc.Prof. Dipl.-Ing. Dr. Klaus Witrisal
Dipl.-Ing. Dr. tech. Ulrich Mühlmann

Assessor:
Assoc.Prof. Dipl.-Ing. Dr. Klaus Witrisal

Graz, September 11, 2018

Acknowledgement

I would like to seize this opportunity to express my gratitude to the people who helped shape this thesis: Prof. Klaus Witrisal for his invaluable support and advice over the entire course of the thesis' work, and Dr. Ulrich Mühlmann for his dedication in helping me grasp the challenging new world of RFID technology and for familiarizing me with the resources and infrastructure provided by NXP. All of our meetings motivated me to progress expeditiously, with their joint professional guidance steering me in the right direction.

I am deeply grateful for the support my family has given me over the course of my studies. My father deserves special credit for investing a decent amount of his time in proofreading. Most importantly, I wish to thank my loving girlfriend Athena for her everlasting wisdom and inspiration.

Abstract

As transmitting information by means of wireless digital communication grows more and more demanding, new possibilities of optimizing data throughput are being explored. In this specific case, the applicability of *blind channel equalization* – a certain kind of adaptive technique – is examined to improve data transmission in RFID systems. The given RFID application is specified under ISO-14443. Very high bitrate transmission (VHBR) from card to reader is the main research interest. Classical adaptive equalizers are impractical in this context, as they require training sequences, whereas blind equalizers do not. A sophisticated RFID system analysis reveals significant properties about coupling behavior and pole-zero dynamics. Based on these insights, a discrete-time baseband system model is derived for general equalizer testing. After a review of potential equalization methods and their attributes, a new blind equalization algorithm is proposed. A general modification, also suitable for other blind equalizers, is contributed in this thesis: It resolves issues of undesired local minima caused by using real-valued constellations (like BPSK) in combination with complex-valued channels and/or equalizers. Its validity is verified by simulation. The performance of the new equalizer (wNCMA) is evaluated in terms of bit error rate (BER) and compared to the MMSE equalizer (which is theoretically optimal, but impractical under ISO-14443, and thus used as a simulation benchmark only). The results place the wNCMA's performance close to the MMSE's for a wide range of system coupling values. Propositions to resolve expected practical issues are made. Simulations have been implemented in MATLAB. In summary, this thesis offers promising new equalization methods for improving RFID technology at VHBR.

Tags: Digital Communication, Adaptive Channel Equalization, Blind Deconvolution, CMA, RFID, VHBR, ISO-14443, Baseband Channel Modeling, MLS, IRS

Contents

1

Introduction

"A good engineer is always a wee bit conservative, at least on paper."
– Montgomery Scott

Technology nowadays is advancing faster than ever before – and communication plays a vital role, both as an enabler and a hallmark of this progress. Communication around the globe, immediate and at any time, has become a "natural" part of everyday life. Beyond inter-human communication, electronic technologies are used increasingly to enable all kinds of devices to interact and exchange information with others of their kind. Our surroundings thus are becoming more interactive, more intelligent, often more pleasant to live in, as communication technology makes many of our daily activities less time-consuming and more effective.
A prominent example of such technologies are contactless smart cards or tags. These are objects small enough to be carried around (like common credit cards) or to be attached to cargo containers in order to have logistics processes run more smoothly and efficiently. Contactless payment is made possible by specially secured bank cards, reducing the transaction time at the checkout counter to less than five seconds. Similar cards are used to grant access to public transportation "in no time".
Although various products are readily available and applied in numerous fields (pet tracking and identification, measures against shoplifting / car theft, security for building access, control of industrial processes, medical applications), this technology area is still young and full of opportunities in research and functional improvement. Is it possible to tweak this technology so that it becomes even faster, smaller, safer, more robust, less expensive...?[1]

This thesis explores the just presented question by combining *RFID* (Radio-Frequency Identification) technology with advanced digital signal processing techniques subsumed under the name *Blind Equalization*. The focus lies in speed: RFID communication, as defined in the ISO/IEC-14443 standard, includes data rates up to 848 kbits/s. Amendments to define higher bitrates (called VHBR or VHDR = Very High Bit/Data Rates) are still a matter of open discussion. System properties such as inductive channel coupling have a significant impact when it comes to VHBR transmission. Special measures are called for in order to eliminate or at least reduce the impact of interference and thus to facilitate error-free transmission of information. This is where channel equalizers surface. The chosen approach of *blind* equalization – and the advantages and restrictions that come with it – are described prior to the hypothesis formulation (Section 1.2) and examined in detail in Chapter 3.

Before the hypothesis can be stated, a short but relevant context about RFID technology is given, followed by a brief explanation of what "blind" means in conjunction with equalization.

---

[1] Keeping in mind fears and concerns about data abuse and total supervision, research in this field is a delicate matter. There lies a great responsibility in making sure all data is protected from unauthorized access, and in educating the public what can or cannot be done with RFID technology. The scope of this thesis within ISO-14443 norm is not fit for total surveillance (because the range is less than 10 cm), and accomplishing higher datarates by equalization may even benefit encryption.

1.1 Context

The information given below is based on [1], in which additional as well as more detailed information about the topic is provided.

1.1.1 RFID - Radio Frequency Identification

RFID technology is usually based on the physical principle of *inductive coupling*. When electric current flows through an inductor (coil) L_1, a magnetic field builds up in its vicinity. If the current changes over time, the magnetic field also changes according to the laws discovered by Faraday and Maxwell. Now, if there is a second inductor L_2 close to the first one, the presence of the changing magnetic field will induce a current in the second inductor, which in turn produces its own magnetic field. The two fields merge, and the interaction of L_1 and L_2 is then called *mutual inductance*:

$$M = k\sqrt{L_1 L_2} \tag{1.1}$$

The parameter k indicates how strong or weak the coupling is, it is a relative measure ($k \in [0, 1]$). Value 0 means L_1 and L_2 are not interacting at all (either due to infinite distance or shielding). At value 1, both inductors possess an identical magnetic flux; this special case is the basis for the ideal transformer. In RFID applications k may approach 1 if the card touches the reader antenna, but values lower than 1 are the common case. Practical couplings may range down to 0.01 or even lower.

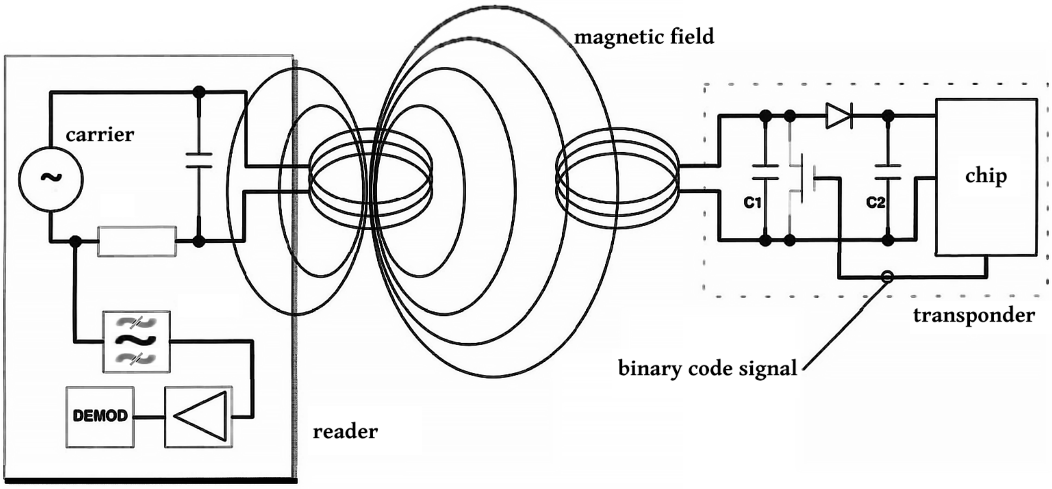


Figure 1.1: Load modulation process between RFID reader and transponder (card). Direct copy of [1], translated to English

As long as coupling is just sufficient, energy can be exchanged between the two inductors. The chip on the card – or similar object, "card" in the following – uses the voltage induced by the reader as its own power supply. The transponder is then called "passive". "Active" transponders include a (miniature) battery as chip power supply, but still require the interactive power of the reader to be able to transmit its response.

RFID devices use the *magnetic* field as communication channel. The reader device generates a carrier frequency, for example a 13.56 MHz sine wave. This electrical wave causes a current flow in the reader antenna – regularly changing its flow direction, thus producing an electromagnetic field. The wavelength of $f_{carrier} = 13.56\,\text{MHz}$ is $\lambda_{carrier} = 22.12\,\text{m}$, which means transmission takes place in non-radiative (reactive) near-field[2], where the magnetic field component is clearly dominant. Beyond the near-field, inductive coupling is not possible: the radiating electromagnetic waves have no retroactive inductive effect on what happens in the near-field. This communication strategy is thus inherently resilient to EM-interference from the outside.

The reader detects if a card enters its field. It then signals the card to establish a communication link and subsequently requests a response signal from the card. The card responds via *load modulation*, which means it changes its own load resistance according to some binary code signal (using the transistor shown in Figure 1.1). This change influences the card's antenna current and thus the overall magnetic field. In terms of communication theory, this load modulation results in the carrier signal being amplitude-modulated according to the card response signal. The reader can detect these (subtle) signal variations at its antenna and – as shown in Figure 1.1 – reconstruct the card message by (bandpass) filtering, amplification, and demodulation procedures.

1.1.2 ISO/IEC 14443: Overview

RFID applications using 13.56 MHz are standardized in the ISO/IEC 14443 norm. It is labled *Identification cards — Contactless integrated circuit cards — Proximity cards*. Proximity indicates operating distances up to 10 cm. The norm currently[3] contains four parts, of which the second is most relevant to this thesis. The contents of the other three are summarized now, following a more detailed description of Part 2.
Part 1: Physical characteristics [2] describes six classes of card dimensions, antenna locations, as well as tolerances, and material and robustness demands (bending, breaking, EM radiation);
Part 3: Initialization and anticollision [3] defines rules for a standardized communication protocol between reader and card (frame format, timing and states);
Part 4: Transmission protocol [4] specifies the activation of the protocol and block transmission.

Part 2: Radio frequency interface [5] defines the range of magnetic field strength (1.5 – 18 [A/m]) and introduces two types (A, B) of communication signal interface. Common to both types is the following quoted requirement (see [5, sec.7]):

> The PCD modulates the amplitude of the alternating magnetic field strength with modulation pulses in order to transmit data from the PCD to the PICC.
> The PICC loads the alternating magnetic field with a modulated subcarrier signal (load modulation) in order to transmit data from the PICC to the PCD.

Bitrate [kbit/s]	f_{sub} [kHz]	Subcarrier periods (T) per bit
106	848	8
212	848	4
424	848	2
848	848	1
1695	1695	1
3390	3390	1
6780	6780	1

Table 1.1: Type A/B: Subcarrier frequency definitions corresp. to bitrates

---

[2] near-field region boundary defined as distance $d < \lambda/2\pi = 0.159\lambda$, here $d < 3.5\,\text{m}$

[3] revised version of 2016

PCD means reader, PICC means card. Relevant for this thesis is the PICC to PCD direction under Type B specification. In initialization phase, every card must respond at rate 106 kbit/s. Afterwards, the card message may be transmitted at higher rates corresponding to $f_{carrier}$- or f_c-fractions of 2 (see first column in Table 1.1). The subcarrier frequency f_{sub} is always at least 848 kHz ($f_c/16$), which means lower bitrates are achieved by stretching the duration of one bit over several subcarrier periods (see first plot in Figure 1.2) Bitrates higher than 848 kHz are labled "VHBR", which stands for *very high bit rate*.

The subcarrier is BPSK (binary phase shift keying) modulated. The bit representation is NRZ-L (non-return-to-zero level), i.e., the bits assume states ±1. According to these states, the subcarrier phase is either at $0\,°$ (+1) or at $180\,°$ (-1). In VHBR, where bitrate and f_{sub} are equal, the modulation reduces to a special form of BPSK called *Manchester coding*. Figure 1.2 shows examples corresponding to Table 1.1.

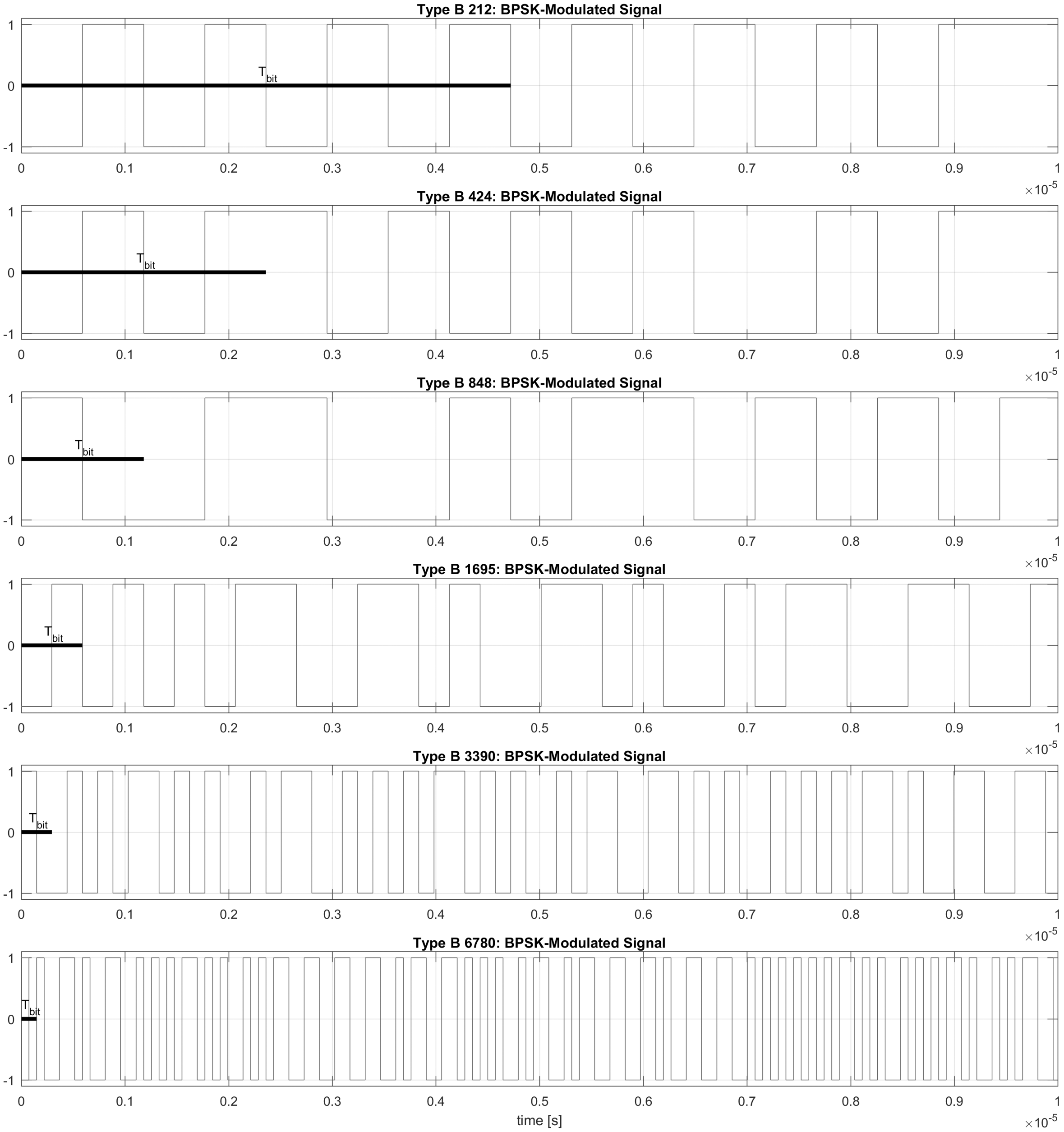


Figure 1.2: Type B BPSK modulated subcarrier signals for several bitrates

Timing is also defined: After a guard time period the subcarrier is generated for at least $80/f_{sub}$ seconds for synchronization. This is equivalent to continuously sending just logical "1" during that time interval. Synchronization ends with a transition from logical "1" to "0" (first occurance of phase change). Then, the message starts.

1.1.3 ISI: Intersymbol Interference

Usually, received message signals do not look like well-behaved square pulses anymore. Traveling through the "communication channel" distorts them. One effect of this distortion is called ISI: A pulse is not confined to its own time slot, but instead expands (both directions possible) and influences neighbouring pulses (see Figure 1.3).

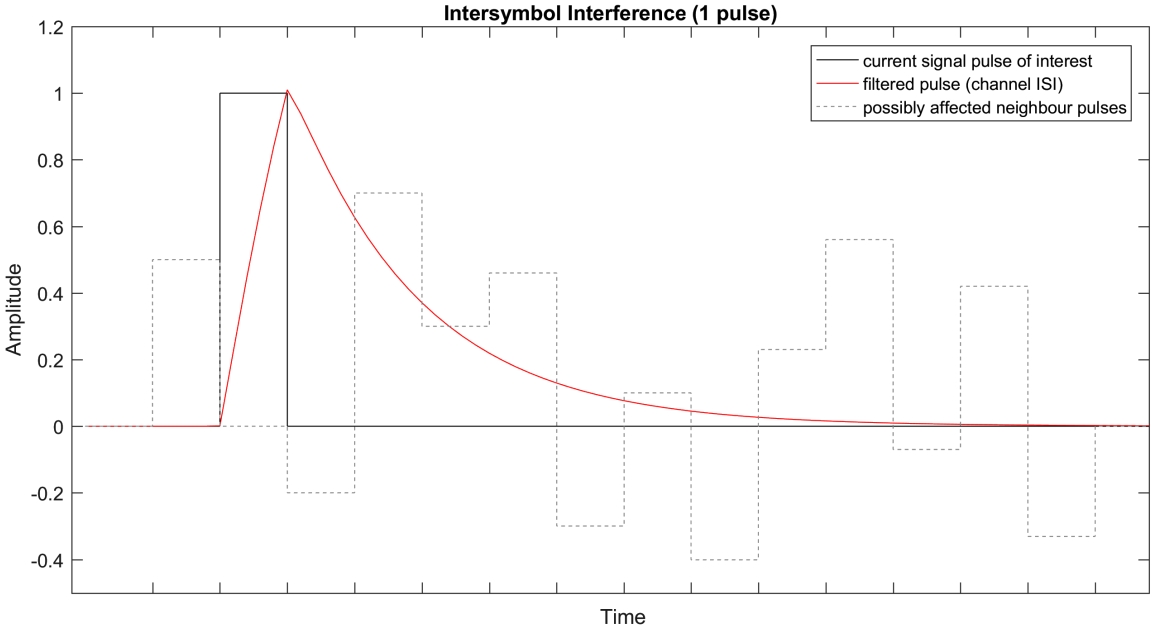


Figure 1.3: The ISI effect

1.1.4 Blind Deconvolution

ISI is undesirable. Equalizers are applied to counter ISI. If such an equalizer is *linear*, and the communication channel is also assumed to be LTI[4], the process of ISI removal is called *deconvolution*. Explanation: The original signal *convolved* with the channel response results in a distorted signal. In order to undo this (to restore the original square pulses), a second convolution with a filter response is applied, which is supposed to offset the effect of the first convolution – and is thus called *deconvolution*. To achieve this, the channel response must be known.

Common adaptive equalizers solve the deconvolution problem as follows: The channel response is not known, but the input message signal is. It is called *training sequence*. No information is transmitted, but now the input and output of the channel are known, so that the channel response can be identified. The procedure is called *system identification*. In a final step, the newly gained knowledge about the channel can be used to design a filter for deconvolution – or if used in real-time, to continuously adapt a filter to a varying channel.

---

[4] linear and time-invariant

The adjective *blind* means that the channel response is *not* known, neither is the input signal, but deconvolution is attempted anyway. At this point, the task becomes much more complex. Proper modelling and statistical assumptions are necessary to – at least partially – compensate for the unknowns. Details on how to do this are given in Chapter 3.

1.2 Complication

The general transmission concept is well defined by the ISO-14443 standard. There are "grey areas", however, which are purposely formulated cursorily and thus allow various approaches for implementation or invite research for improvement. VHBR is such an area. For each standard bitrate up to 848 kbit/s, the subcarrier of 848 kHz is used (see table 1.1). It is spaced just far enough from the carrier to be filtered and demodulated separately, and still close enough to not run into severe phase distortions. And it is quite a static situation, so the system can be adjusted to optimally support the 848 kHz subcarrier.

Not so in VBHR. As can be seen in Table 1.1, the subcarrier changes along with the bitrate. The signal shape is then equal for all VHBR signals (which can be advantageous; see Figure 1.2), but the channel characteristics are completely different for each new subcarrier. Especially phase changes around the carrier are an issue. They are related to the coupling between card and reader (see also Chapter 2). Resonances in the channel are coupling distance-dependent, which means they can appear in a wide area around the carrier frequency – posing a particularly challenging situation in terms of equalization.

The "easy way" of a static filter is clearly not an option. The adaptive method briefly mentioned in Section 1.1.4 seems much better suited, it can track changes in the system, as described in the previous paragraph. Unfortunately, using training sequences in this context is infeasible, because (1) the information rate will significantly decrease if every card response additionally contains the training sequence, (2) activation power and thus transfer times are limited, and most importantly (3) compatibility becomes impossible: none of the already existing cards will have that training sequence on it, since the current ISO-14443 protocol does not exhibit training sequence support.

1.3 Hypothesis

In order to address the challenges outlined in Section 1.2, and building on the inspiration from literature research (see Section 1.4), this thesis explores if RFID communication (card-to-reader transmission) can be improved by

- applying a digital **blind** channel equalizer to the given RFID system
- using insights from system analysis to tune equalizer settings
- using knowledge of input signal properties

and is subject to the following conditions/restrictions:

- compliance with the ISO-14443 standard
- focus on VHBR
- no training sequence
- real-time operability

- ▷ low implementation cost
- ▷ AWGN $\ll$ ISI

More specifically, research shows that the **Constant Modulus Algorithm** (CMA) is a promising candidate for this specific application. The focus of studies will thus be on CMA or similar equalizers. Their convergence behavior is also of interest. Evaluation measures will be based on *bit error rate* (BER) in comparison to the (to this case non-applicable) adaptive MMSE equalizer.

1.4 Present research status

Across the reviewed literature, Yoichi Sato is credited for proposing the first blind equalizer in 1975 [6]. Since then, several other methods have been proposed in [7–11] and have been enthusiastically applied first and foremost to wireless communication systems, specifically for channel equalization, antenna array beamforming, and multi-user detection and separation. Other application areas include underwater acoustics, seismic data deconvolution, and digital image restoration to name a few. Recently, blind methods have also been associated with Kalman filters and neural networks.

The literature review exposed the **Constant Modulus Algorithm** (CMA) as the most commonly used blind method in the context of digital communication. CMA is frequently chosen due to its general robustness when PSK (intended) or even M-PAM / M-QAM (collateral, comp. [12]) modulation schemes are used, and due to its simplicity and low computational cost. It appears in various modified versions, as researchers often go for customizing the basic algorithm to specific use cases. Paper [13] is recommendable for its comprehensive compilation of information about blind equalization in general.

CMA has been applied to RFID, but in an entirely different context. Ultra-high frequency (UHF) RFID is the center of interest in [14], based on the “Gen2” standard proposed by *EPCglobal*[5], operating about just below 1 GHz. Given wavelengths around 30 cm, near-field effects are of no concern. ISI caused by multipath fading is the dominant factor here, since operating distances range from 1 m (minimum!) up to 15 m or more. Communication channel characteristics are thus very different in this case. Otherwise, the general structure of the receiver and HF interface, and the application as equalizer for the received card (tag) signal, show similarities to this thesis. Bit error rate (BER) is used as the main evaluation measure, but there is no reference to other equalizers; the CMA is merely tested against the unequalized channel (it obviously performes better).

System modelling has been done in 13.56 MHz-RFID, but limited to the communication direction from *reader to card*. Research in papers [15, 16] has been encouraged by 2011/2012 amendments in the ISO-14443 standard, which states[6]:

> There is a need for both ASK and PSK methods for all data rates (e.g. fc/8, fc/4, fc/2, 3fc/4 and fc) in order to optimize for local requirements. Worldwide Interoperability shall be achieved by mandating both methods for the PCD and allowing the PICC to adapt to local requirements with different methods. In order to keep the specification consistent both technologies ASK and PSK shall be specified in the same amendment [...]

---

[5] An organisation for standardization (like the ISO), but on a global level and with main interest in global norms for RFID product identification tags

[6] JTC1/SC17/WG8: `http://wg8.de/wg8n1854_Disp_of_Comments_on_FCD_14443-4_FPDAM2_VHBR-ASK.pdf`

Paper [15] states that with the PSK VHBR protocol it is possible to implement training sequences, fostering the application of an adaptive DFE[7] on the card's chip. Paper [16] examines this adaptive equalizer, assumes a linear channel and equalizer, and shows simulation results. Paper [17] uses a nonlinear channel model (also *reader to card* direction) and applies a ternary-state MLSE (Viterbi) method, assuming the channel length is always 4 (taps, coefficients). Equalization in this direction appears to be easier, because the signal is modulated onto the carrier directly (not onto subcarriers as in the opposite direction examined in this thesis).

System modeling in *card to reader* direction has implicitely been done in [18], where a (direction-independent) state-space representation of a typical 13.56 MHz-RFID circuit is derived. Conversion to more specific transfer function models (dirction-dependent) is also shown. Insights of this paper will yield the starting point for the system analysis in Chapter 2.

Studies in the card-to-reader ISO-14443 context have been conducted in [19], which focuses on increasing the transmission data rates by either increasing the number of modulation states (M-PSK) or increasing the subcarrier frequency. ISI equalization is also considered, although both modeling and filtering are done in the *passband*, which leads to sample rates of up to 32-times 13.56 MHz. Evaluated methods include a passive filter (analog), an IIR bandstop filter (digital), a zero-forcing FIR filter (digital, based on model inverse), and adaptive RLS[8] (digital, training sequence). The need for *adaptive* equalization due to changing channel characteristics is duly noted. Blind equalization or baseband filtering approaches are not mentioned.

1.5 Rationale

As a conclusion to the findings and researches so far, the significance accompanying this very specific research topic becomes clear. Apart from the fact that CMA has not been applied or tested as stated in the hypothesis (see Section 1.3), several other reasons motivate the work of this thesis, including:

- Research has shown that CMA has been applied successfully in similar, related areas of communication.
- Investigation of CMA specifically makes sense due to its low implementation cost.
- Analysis of the RFID system can help resolve issues in similar systems or be used to design even better equalizers in the future.

1.6 Roadmap

Chapter 1 set out to introduce RFID technology, arguing for the necessity for equalization. The hypothesis has been stated. Chapter 2 presents a detailed system analysis of a typical RFID schematic, a system identification method to support the analysis, and a system model based on these insights. Chapter 3 presents blind equalization methods, their properties and behavior, and requisite theory in digital communication and signal processing. A new algorithm designed for the given RFID system is proposed, based on a mixture of existing algorithms and contributing new modifications. Chapter 4 presents details of the implementation and shows performance results. Chapter 5 summarizes and concludes the thesis.

---

[7] decision feedback equalizer

[8] recursive least-squares

2

System Analysis and Modelling

Communication systems are usually composed of various processing steps, forming a chain in which the source data is repeatedly "transformed" into different signal representations until it arrives at the receiving end (see Figure 2.1). Errors occur during transmission over the *channel*. Source and channel coding intend to make the transmission most efficient as a trade-off between information density and redundancy (added e.g., for bit error detection, error robustness towards specific channel types, or encryption).

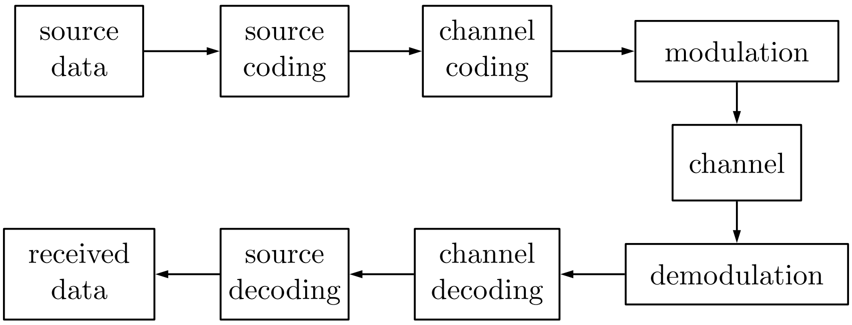


Figure 2.1: Block diagram of a common data transmission process

However, before addressing any notion of efficiency or general improvement, in this chapter we take a look at what creates those errors in the first place. Specifically the right part of Figure 2.1 will be examined, containing modulation, the channel, and demodulation. These three blocks in the chain are defined by the underyling physical RFID circuit, which explicitly represents the high-frequency behaviour (meaning in the *passband*). This chapter will arrive at an equivalent representation combining these three blocks in one block. This form of system representation is more convenient for equalizer evaluation; it is called *equivalent baseband representation.* The intention is to pass a binary message **input** signal through this equivalent **system** (by convolution → filtering) so that the received **output** signal is the (already demodulated) message – distorted by the system (see also Figure 3.1 for an illustration). By this approach, the system's influence on the message can be examined in a most distinct way.

Starting with the analysis of the representative electric circuit of the RFID system (section 2.1), which is given as a *state space model* (see Section 2.2.2), fundamental insights are attained that are necessary to proceed from this rigorous state space model to a reduced model adequate for this thesis' purposes. Section 2.2 describes the process of deriving the equivalent baseband model (still based on the state space model) and implies the importance of complex numbers in this context. In Section 2.2.4, the state space approach reaches its limits regarding its intended baseband conversion validity. As a consequence, the system needs to be reexamined from a *second perspective*: System identification is applied according to the just mentioned "input—system—output" principle using the MLS and IRS methods (Section 2.3). Similarities and differences to the original circuit model are established. Finally, a simplified discrete-time baseband model will be constructed for equalizer testing (Section 2.4), which is based on the system identification results.

2.1 HF Circuit Analysis

The basics of RFID transmission have been summarized in Section 1.1.1. Elements of the system in Figure 1.1 will be presented below in the form of an electrical network schematic. The analysis of electrical networks is a mathmatically well established means of determining all currents and voltages at each component. It is a very detailed and exact way of describing the system in continuous time.

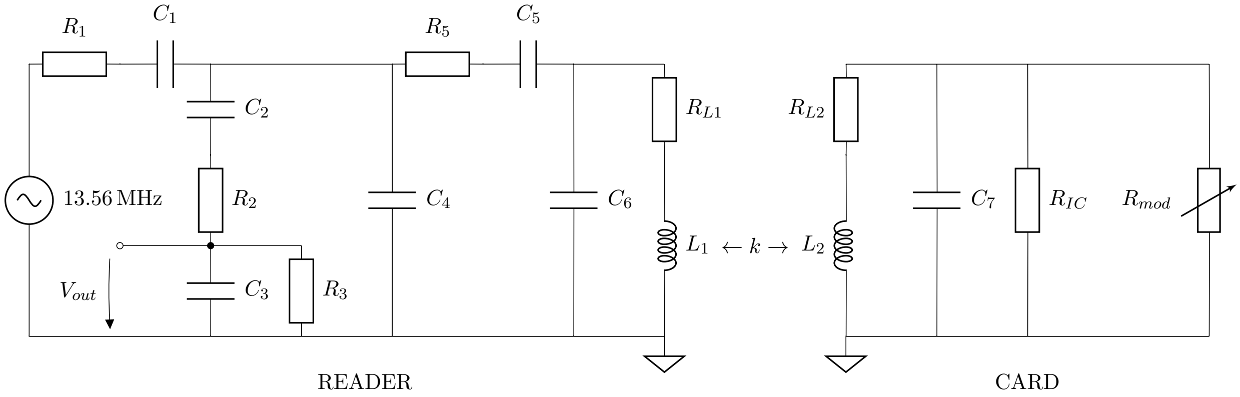


Figure 2.2: RFID circuit schematic

Figure 2.2 shows the RFID circuit in consideration as a linear system. Linear implies that it is built of well-behaved components, their interaction is then defined by a system of simple differential equations. These equations can be found by nodal or mesh analysis, using the laws of Kirchhoff, the principle of superposition, and the component-specific equations. Another advantage of linear networks is that one (or a group of) component(s) may be exchanged by mathmatically equivalent components so that certain features of the circuit can be analysed more easily (comp. Norton/Thévenin theorems). Advanced (active) components can occasionally be represented by linear elements as well.[9]

The variable resistor R_{mod} is such a substitution. In Figure 1.1 the transponder card contained a chip (integrated circuit, IC) and a transistor. These (nonlinear) elements have been replaced by two resistors (R_{IC}, R_{mod}), which sufficiently model the system's behaviour within the frequency range of interest (13.56 MHz). R_{IC} represents the constant resistive load of the integrated circuit as it will affect the reader's side. In parallel, R_{mod} models the switching behaviour of the transistor[10]. Here, the expression "load modulation" becomes evident once again: The card changes its own internal resistance – a quantity the reader identifies as a *load* – and by changing it in a defined way, for example,

- $R_{mod} = 1k\Omega \quad \rightarrow$ logic *low* state (0)
- $R_{mod} = 0.5\Omega \quad \rightarrow$ logic *high* state (1),

the reader can detect and demodulate these load changes of the card. As the communication direction from *card to reader* is examined in this thesis, these binary state changes are considered to be the **input signal** to the system. However, in order to explain the circuit properly, the process that leads to the appearance of the input signal needs to be outlined first.

---

[9] at least in a simplified context, as an approximation

[10] taking *out* of account the transient behaviour, assuming transition speed is orders of magnitude faster than $T_s = 1/f_s = 1/f_{carrier}$

2.1.1 Time domain behaviour

It starts with the voltage source on the left end of the schematic, which generates the sinusoidal carrier frequency of 13.56 MHz. This is quite important to note, because with this source deactivated the entire circuit is *off*. Reader and card are completely inactive, no communication is possible. But as soon as the carrier is activated a current – oscillating at 13.56 MHz – flows through every network element – most importantly through the reader's antenna L_1 (inductor). In consequence, a magnetic field is generated, fulfilling the equation (comp. [20, p.347]):

$$\frac{d\Phi(t)}{dt} = L\frac{di(t)}{dt} = v(t). \tag{2.1}$$

The magnetic flux Φ follows the change of current over time. The flux is related to the field by (see [21, p.45ff])

$$d\Phi = \mathbf{B} \cdot d\mathbf{S} \qquad \Rightarrow \qquad \Phi(t) = \int_{\mathbf{S}} \mathbf{B} \cdot d\mathbf{S} \tag{2.2}$$

where the generic reference surface $\mathbf{S}$ (i.e., the surface spanned by the antenna) is divided into infinitesimal surface elements $d\mathbf{S}$. Variations in flux and field (over time) are considered at each of these tiny segments separately. The field $\mathbf{B}$ will thus also oscillate at 13.56 MHz, since the surface is constant. As a side-effect, an electric field $\mathbf{E}$ additionally emerges due to the Maxwell-Faraday Equation (2.3, see [21, p.49f.])); however, it only has an effect in the far field ($d > 3.5\,\mathrm{m}$ in this case, see Section 1.1), yet it will still radiate electro-magnetic waves. For this reason, the ISO standard [5] strictly limits the field strength.

$$-\frac{\delta \mathbf{B}}{\delta t} = \nabla \times \mathbf{E} \tag{2.3}$$

If the card's antenna L_2 is in range, both system parts become coupled inductively. The magnetic field $\mathbf{B}$ is "attracted" by the arriving antenna, the flux induces the oscillating current of L_1 into the card's circuit at L_2. To quantify the interaction of the two system parts, the coupling factor k is defined as:

$$k = \frac{M}{\sqrt{L_1 L_2}} \qquad k \in [0,\ 1] \tag{2.4}$$

Since an electric current now flows through the card's circuit, the card can draw power from it. The internal chip is activated. To boost the energy transfer, the *resonance principle* is used: Adding capacitances with adequate values tunes both circuit parts to have a common resonance frequency (details about frequency behaviour can be found in next Section 2.1.2):

$$\omega_{res} = \frac{1}{\sqrt{LC}} \qquad \Longleftrightarrow \qquad f_{res} = \frac{1}{2\pi\sqrt{LC}} \tag{2.5}$$

If this frequency is dimensioned to be equal (or close) to the generated carrier frequency, the oscillating energy is entirely "caught" between inductor and capacitor. In case of the reader circuit, the significant two elements form a series resonance (mainly L_1, C_5), the effect of which is explained in the following: Taking turns, the inductor stores energy into its magnetic field until no current flows through it anymore. The voltage reaches a maximum (that may well be magnitudes higher than the one generated by the voltage source). Then the capacitor – due to the high voltage – starts building up an electric field between its plates → the capacitor's way of storing energy. As the electric field grows, current will again flow and as a result the voltage decreases until the electric field is complete – and the inductor's magnetic field vanished. At this stage, all energy is stored in the capacitor. A maximal current now flows through the circuit (also magnitudes higher, like the voltage), causing the inductor to build up its magnetic field again. This recurring energy shift happens at 13.56 MHz. If both card and reader are tuned to this frequency, the resonance greatly benefits both coupling range and energy transfer to the card.

The card's circuit is also a resonant one, but it forms a parallel resonance including L_2, C_7, and both R_{IC} and R_{mod} (see Figure 2.2). As a result, the card's impedance will be at its *maximum* at 13.56 MHz. In turn this means the card acts as a significant load, thus is more likely to consume the energy provided by the reader.

Currents and voltages can become uncontrollably high in resonant networks. Hardware may easily be destroyed as a consequence. Protection measures are necessary: The chip (IC) on the card should be able to compensate possibly harmful currents. So besides changing its own internal resistance to modulate a message, the IC needs to change its own resistance to absorb sufficient supply current (or identically, keep the operating voltage steady around $3 - 5\,\mathrm{V}$), but beyond that *limit* any extra current via a (variable) shunt resistor. This functionality is also adequately represented by the two resistors R_{IC} and R_{mod}.

Now the card modulates the binary response signal by switching the resistance of R_{mod}. The switching is done in synchronization with the carrier frequency of 13.56 MHz. The chip on the card can produce clock signals that correspond to fractions of 2 of the carrier frequency. Out of these signals, the subcarrier (switching) frequency is obtained. Possible frequencies have been shown in Table 1.1. The message is encoded by changing the phase of the subcarrier according to a given bit sequence (compare Figure 1.2). Because of the synchonization the switching ideally happens during the zero-crossing of the carrier frequency. When measuring the oscillating current anywhere in the circuit, the visible effect will be a small amplitude modulation in the carrier waveform (bottom trace of Figure 2.3). Amplitude limits of the modulation for each card class are given in the ISO-14443-2 standard [5] in addition to the minimum modulation level a reader must be capable to deal with. These limits depend on the magnetic field strength. The card's circuitry is required to adapt its internal impedance so that the load modulation amplitude (see ISO 10373-6) of the response signal is always only slightly above the reader's detection threshold.

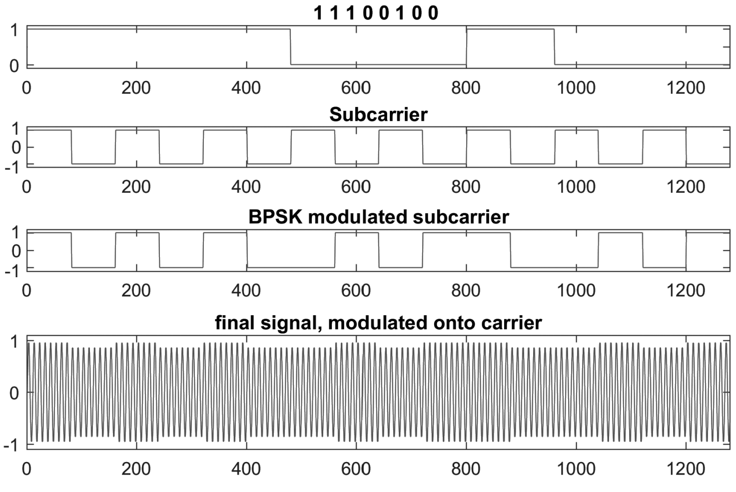


Figure 2.3: Load modulation process of card signal by subcarrier phase shift, $f_{sub} = 848\,kHz$

The remaining network of resistors and capacitors in the reader has a filtering effect on the electrical signal. The parallel capacitors C_4 and C_6 are used for impedance matching in conjunction with C_5 (forming a Π-network with overall capacitance C_{sum}), which intensifies the resonance at the reader's antenna. At the node between R_2 and C_3 the voltage (V_{out}) is tapped and sent to post-processing units (more filters, demodulator, DSP or FPGA hardware) to reconstruct the card's bit sequence.

2.1.2 Frequency domain behaviour

The interaction of resonance and inductive coupling is best comprehensible in frequency domain, i.e, by analyzing the frequency spectrum. The following explanation intends to show the movement of the resonant peaks in the spectra of Figures 2.4, 2.5 and 2.6. The key events of this movement are shown out of the perspective of the variable coupling factor k.

When infinitely apart ($k = 0$), both reader and card have their separate resonance frequencies. Each individual circuit is tuned to $f = 13.56\,\text{MHz}$ according to (2.5). Strict equality is practically unrealistic but a sensible starting point for this explanation:

$$\omega_{res,1} = \frac{1}{\sqrt{L_1 C_{sum}}} \qquad \overset{!}{=} \qquad \omega_{res,2} = \frac{1}{\sqrt{L_2 C_7}} \tag{2.6}$$

As the card approaches the reader (e.g., $k = 0.001$, see Figure 2.4), the coupling is not strong yet ("loosely coupled"), so the individual resonances dominate over the coupling effect. However, the presence of the source-generated carrier current/voltage amplifies this coupling effect. So in spite of a low coupling factor k an energy transfer to the card is possible. On the downside, as the coupling factor stays nominally low, a change in the card's resistance (R_{mod}) will be hardly noticable on the reader's side (reader-side load will be indistinguishibly small, regardless of the value of R_{mod}). A remark: the bandwidth in which the resonant amplification effect occurs is very narrow. If the subcarrier frequency is high (as in VHBR, where the frequency peak would be far away from the dotted line), the resonance will exhibit an opposite effect and instead attenuate the modulated subcarrier signal response (e.g., if a subcarrier was to lie at 5 MHz in Figure 2.4, it would be attenuated by 10 dB).

As the coupling grows stronger, the influence of R_{mod} becomes evident. Looking at the black dotted graph first (in Figure 2.5), there is a slight movement to higher frequencies. The movement continues (in Figure 2.6) with increasing coupling factor k. The card is a parallel RLC circuit. If the resistance (in case of the black curve: $R_{mod} = 1.3\Omega$) is small, the quality Q of the parallel circuit is low. At resonance, L_2 and C_7 are equivalent to an open circuit, so the load detected by the reader circuit merely consists of R_{mod}. The series resonance on the reader's side remains dominant due to the low R_{mod} of the card, but the increasing load on L_1 (due to increasing k) gradually increases the single resonance in frequency.

Looking at the gray graph of Figure 2.5, the bandwidth of the resonance is higher. This phenonenon is called "critical coupling", and is explainable only by taking into account the coupling factor k *and* the presence of the resonance frequency (carrier source active)[11]. R_{mod} increases to $2k\Omega$, thus the quality Q of the parallel resonant circuit increases:

$$Q_{parallel} = R \cdot \sqrt{\tfrac{C}{L}} \tag{2.7}$$

---

[11] The phenomenon of under-critical, critial and over-critial coupling occurs exclusively in tuned (resonant) inductively coupled networks; see [22] for further details.

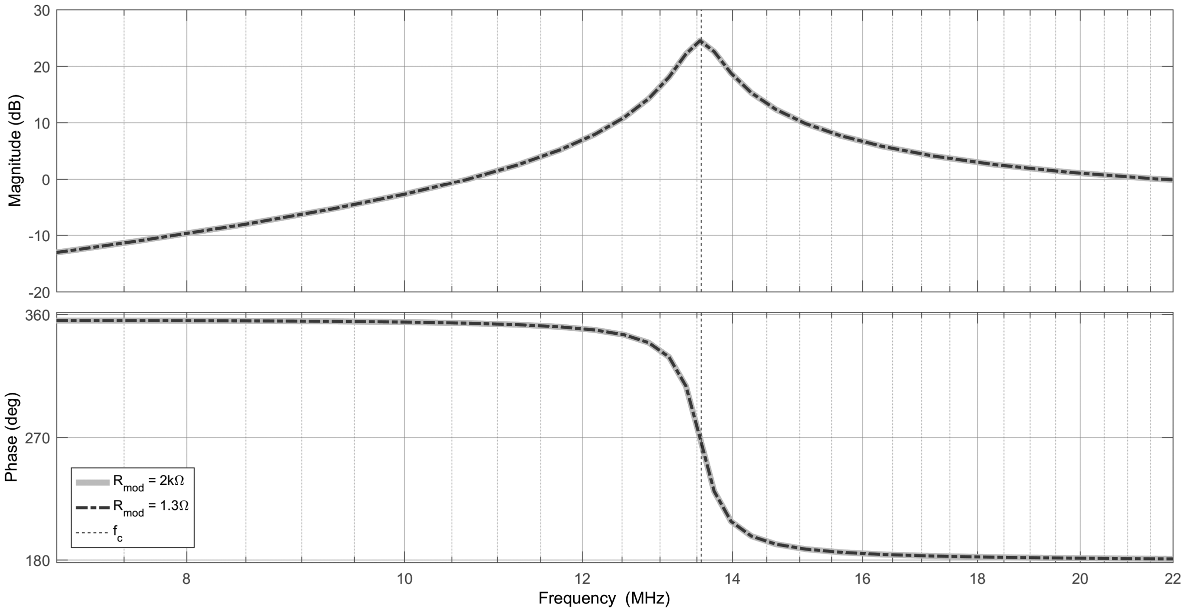


Figure 2.4: Bode Diagram: V_{out} resonances for $k = 0.001$

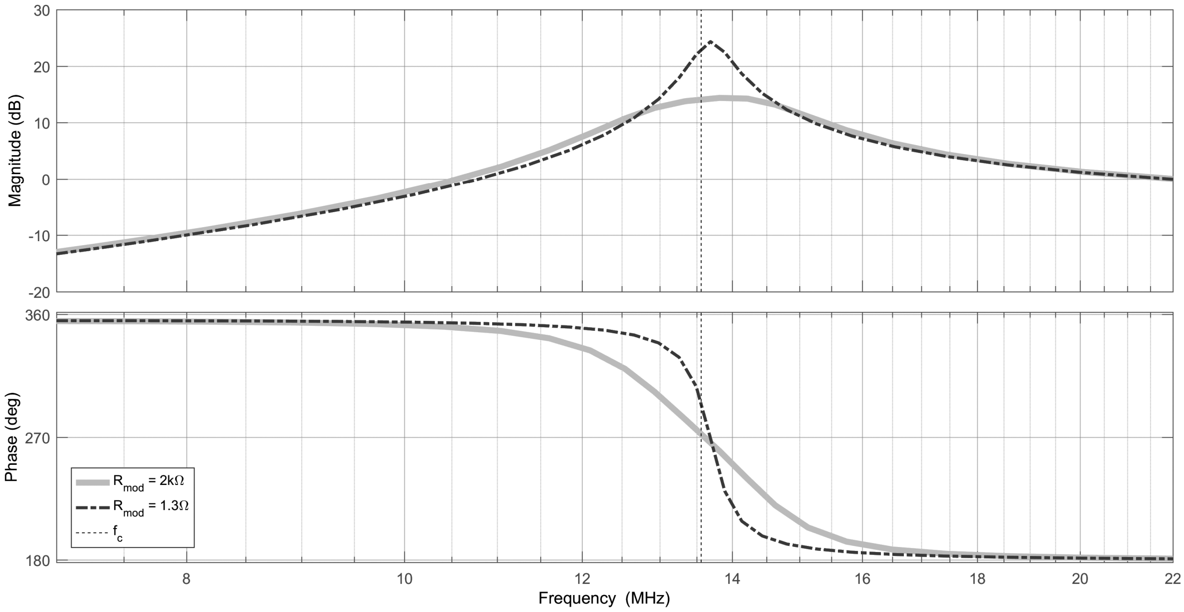


Figure 2.5: Bode Diagram: V_{out} resonances for $k = 0.15$

Critical coupling is defined by both Q and k. In non-resonant coupled systems, the optimal coupling is $k = 1$. In resonant systems, the optimal (critical) coupling is defined as $kQ = 1$, so that a high quality resonance may reach the optimum while k is still significantly lower than 1. In the frequency magnitude spectrum, critical coupling shows a maximally flat response (resonance bandwidth and gain are both maximal then).

However, as the critical coupling is already achieved at a low k, further increasing k at this point results in what is called "over-critical coupling". It is defined as $kQ > 1$ and the effect can be seen in Figure 2.6: The formerly optimal broad bandwidth falls apart and splits up into two separate resonance frequencies, one moving towards lower frequencies, the other towards higher frequencies. An increase in k moves both resonances farther away from the original res-

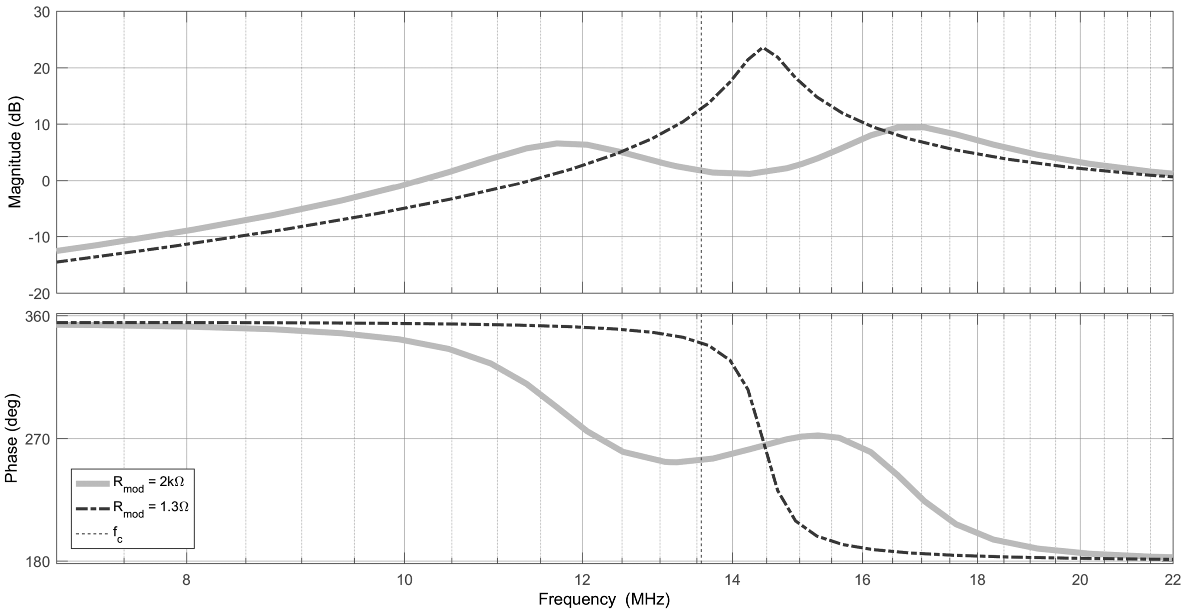


Figure 2.6: Bode Diagram: V_{out} resonances for $k = 0.35$

onance frequency (f_c, dashed vertical line). They have different magnitudes; this is mainly due to the difference of inductance between L_1 and L_2[12]. As can be seen in the overcoupled case, a load modulation via subcarrier benefits from a generally wider spectrum bandwidth. The two resonances are, however, not at all guaranteed to have a positive, amplifying effect, and their positions vary over an extensive frequency range with varying k. Additionally, the responses for the two cases of R_{mod} are now most dissimilar, especially their phase responses. When starting to switch R_{mod} for message transmission, severe phase distortions will be the result.

At this point it becomes obvious that equalization does make sense. In the case of low coupling, the resonance is sharp, which corresponds to a narrow bandwidth. While positively affecting the energy supply, the resonance causes oscillations ("ripple") in the time-domain signal (the message signal), resulting in severe ISI at that end. In the case of high coupling, the oscillating tails grow shorter (corresponding to a "larger" bandwidth → less ISI), but phase variations disturb the transmission now.

Equalization in the passband is costly. If analogue circuitry is to be used, many additional, low-tolerance components will be required; complexity grows and energy efficiency becomes worse. If digital hardware (DSP/FPGA) were to be used, very high sampling rates would be required if the equalizer was to be applied in the passband directly. It is thus common practice to apply the equalizer in *baseband* – meaning *after* demodulation. Sampling frequecies may be chosen much lower, so as to sufficiently sample the subcarrier signal instead.

The following section describes how the analyzed high-frequency network can be translated to a baseband representation without losing information about its behaviour. This representation is therefore denoted as "equivalent" baseband.

---

[12] due to antenna dimension, and windings "N"

2.2 Conversion to Equivalent Baseband

Section 2.2.1 gives an overview of down- and upconversion and necessary related background. More details can be found in [23,24]. The basic form of the system model – relevant for Chapter 3 – is derived. The following Section 2.2.2 then describes the actual conversion; state space model and transfer function details are refered to in [24, 25].

2.2.1 Conversion principle and objective

The starting point of the translation is the physical signal. It is located around the carrier frequency of $f_c = 13.56\,\mathrm{MHz}$. Since there is a sharp resonance of the system, any frequency content outside this resonance region is attenuated. Because of its bandpass characteristic, a signal like this is called the **passband** signal. It is disturbed by the passband *system*, which is also having its main effect in the very same frequency region. This is the actual physical process, so these signals and systems are *real-valued*. The situation is easily manageable in terms of time-frequency relations, for the Fourier transform offers a benign symmetry (comp. [24, p.47ff., 56f.]):

- The Fourier transform of a real even function is real.
- The Fourier transform of a real odd function is imaginary.
- The Fourier transform of an even function is even.
- The Fourier transform of an odd function is odd.

In summary, the four theorems imply an overall Hermitian symmetry, meaning $g(x) = g^*(-x)$, where the * signifies the complex conjugation. This is important to note, because the real signal or system is composed of the passband spectrum at *positive* frequencies around $+f_c$ (as shown in Figures 2.4, 2.5 and 2.6, blue curve in Figure 2.7), and also around the *negative* carrier of $-f_c$. The rules of symmetry will remain valid, even if the following steps are taken to arrive at the equivalent baseband representation.

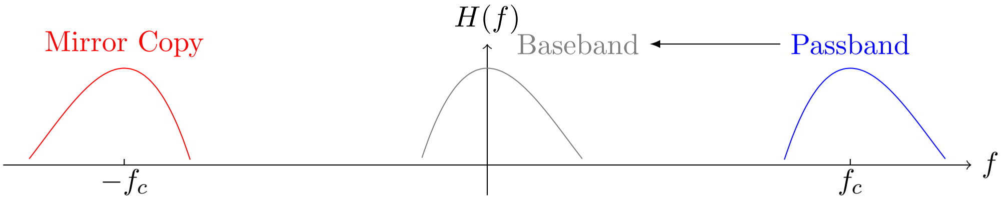


Figure 2.7: Fourier spectrum symmetry for real-valued signals and systems

1. Remove the mirror copy at $-f_c$
2. Shift the passband at f_c to baseband so that it centers around $f = 0$

The order of steps is interchangeable. If the shift is done first, the mirror copy can still be removed (only now it is located at $-2f_c$ instead). Besides, the process is reversible: The **baseband** representation can be shifted back to f_c and a mirror copy set up at $-f_c$ without any loss of information (mathmatically). The procedures are denoted **downconversion** and **upconversion** respectively.

Figure 2.8 visualizes both procedures as signal flow graphs. The signal $s(t)$ is the real-valued passband signal, $\tilde{s}(t)$ is the *complex-valued* baseband equivalent. $P(f)$ keeps only positive frequencies and sets all content at negative frequencies to 0. The operation is also known as *Hilbert*

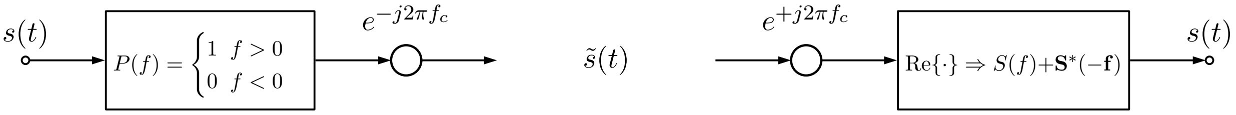


Figure 2.8: Signal flow concept of downconversion (left) and upconversion (right)

transform. After this block (and before the multiplier), the signal $s(t)$ is still in the passband, but has no negative frequency content and is thus non-symmetric in frequency domain – and due to Fourier's theorems complex-valued in time domain. In this state, the signal is called "analytic" signal. The multiplication by the complex exponential $e^{-j2\pi f_c}$ shifts the analytic signal down to baseband. This operation can be interpreted as a demodulation-like process, yet it contains more information than an actual demodulated signal (which would be just real-valued). The resulting signal is, however, similar to a demodulated signal in several aspects, for this reason it is often called "signal envelope".

When the envelope signal is written down in Cartesian coordinates $(a + jb)$, the two parts are named *in-phase* and *quadrature* components in the context of communications. This notation will be important in Chapter 3, but it is convenient to be mentioned right here within the consistent background description:

$$\tilde{s}(t) = s_I(t) + js_Q(t) \qquad s_I(t) = \mathrm{Re}\{\tilde{s}(t)\} \qquad s_Q(t) = \mathrm{Im}\{\tilde{s}(t)\} \tag{2.8}$$

In reverse (see right figure 2.8), the envelope signal is shifted up again by multiplication with $e^{+j2\pi f_c}$, and the negative frequency content can be restored by either taking the real part of the analytic signal (which generically restores the mirror copy) or (equivalently) by adding the mirror copy manually $(+\mathbf{S}^*(-\mathbf{f}))$ or by using Hilbert transform once again. The concept of upconversion is the mathematical basis for a modulation, the "exact" solution which practical modulators are implemented on. The same is true for downconversion and demodulators.

As this chapter is really about systems and not about signals, the downconversion will consequently be applied to systems, the previously analyzed system especially. This is permissible, because from the signal processing point of view, the terms "signal" and "system" are interchangeable – because the convolution operator $(*)$ is commutative [26, p.31]. The disturbance caused by the physical high-frequency (passband) system can be modelled as a filter, applied to the (message) signal $s(t)$ by convolution:

$$s_{dist}(t) = h_{passband}(t) * s(t) \qquad all \in \mathbb{R} \tag{2.9}$$

The filtering shall take place in baseband instead. The filter $h_{baseband}(t)$ is the downconverted version of $h_{passband}(t)$, thus the conversion is already included in the filter response. Obviously this filter response is complex-valued and will usually not exhibit any symmetry. This fact complicates the analysis and also the equalization later on, because many implemented analysis and processing tools are based on these symmetry assumptions and do not support more general (complex-valued) cases of applicattion. But the advantage of this generalization is that everything can be accomplished at low frequencies: The message signals $b(t)$ need not be modulated onto a high frequency carrier, instead it can be directly passed through the equivalent baseband system model which already takes into account the effect of modulation and the distortion at high frequencies. The output is then a baseband representation of the received signal, denoted $\tilde{r}(t)$.

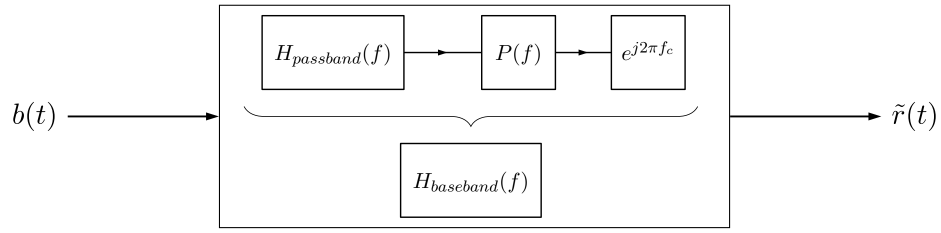


Figure 2.9: Signal flow graph in the equivalent baseband

$$\tilde{r}_{dist}(t) = h_{baseband}(t) * b(t) \qquad \tilde{s},\ h \in \mathbb{C}, \quad b \in \mathbb{R}\ \ or\ \ \mathbb{C} \tag{2.10}$$

The resultig flow graph as shown in Figure 2.9 yields a simplified structure. Only one compact system $H_{baseband}(f)$ remains. The following sections of this chapter deal with finding and quantifying such a system for the given RFID circuit. Once found, Chapter 3 will explore the possibilities to equalize baseband systems of this kind.

2.2.2 Downconversion in Laplace and Fourier domain

The circuit in Figure 2.2 has been provided as a state space model for this thesis. As is evident from the figure, seven capacitors and two inductors form a ninth-order state space model of the general form:

$$\begin{aligned} \frac{d\boldsymbol{x}(t)}{dt} &= \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t) \\ y(t) &= \boldsymbol{C}\boldsymbol{x}(t) + \boldsymbol{D}\boldsymbol{u}(t) \end{aligned} \tag{2.11}$$

The state variables forming vector $\boldsymbol{x}(t)$ are the voltages or currents at each of the nine energy-storing circuit components. That way, the state of the entire network at any given time instant can be computed by solving the set of differential equations according to (2.11). Matrix $\boldsymbol{A}$ is the system matrix primarily containing information about how fast (time-unit: s^{-1}) voltage or current changes occur at each component. The input signal vector $\boldsymbol{u}(t)$ is weighted by the input matrix $\boldsymbol{B}$ depending on where in the circuit the input is to be initiated. Output $y(t)$ is a (scalar) time signal, defined by the state $\boldsymbol{x}(t)$ weighted by the output matrix $\boldsymbol{C}$ and occasionally by a feed-through $\boldsymbol{D}$ of the input (control) signal $\boldsymbol{u}(t)$.

By choosing the states of interest for the input (via $\boldsymbol{B}$) and the output respectively (via $\boldsymbol{C}$), the general model can be reduced to a single-input-single-output state space system[13], which in turn can be converted into a transfer function (that fully characterizes the relation between input and output). This has been done for the chosen output node of V_{out} (see Figure 2.2), the carrier source as tentative input, and keeping *two* reference systems – for both switch values of R_{mod}, which have already been shown during the analysis in Figures 2.4, 2.5 and 2.6. The transfer function $G(s)$ (see [25, p.49]) is defined in the Laplace (frequency) domain:

$$G(s) = \frac{Y(s)}{U(s)}\bigg|_{\boldsymbol{x_0}=0} = \frac{\mathcal{L}\{y(t)\}}{\mathcal{L}\{u(t)\}}\bigg|_{\boldsymbol{x_0}=0} = \boldsymbol{c}^T(s\boldsymbol{E} - \boldsymbol{A})^{-1}\boldsymbol{b} + d \tag{2.12}$$

---

[13] $\boldsymbol{B}$ and $\boldsymbol{C}$ become vectors instead of matrices, and $u(t)$ and $\boldsymbol{D}$ are scalar like the output $y(t)$

It is closely related to the frequency space of the Fourier transform, in fact, the Laplace domain is the more general one, because its frequency variable $s = \sigma + j\omega$ is a complex variable, reducing to purely imaginary $s = j\omega$ in the case of Fourier. The Fourier representation is sufficient if only the steady state of the system is of interest. In this thesis, however, the fast switching of R_{mod} dictates the necessity of transient analysis. In order to shift the transfer function $G(s)$ from passband to baseband, the Laplace transform property of frequency shift is used. Since the shift variable α is purely imaginary, this applies equally to the Fourier transform (comp. [24, p.33,65]):

$$g(t) \cdot e^{\alpha t} \circ\!\!-\!\!\!-\!\!\bullet\, G(s-\alpha) \quad with \quad \alpha \in \mathbb{C} \quad and \quad \mathrm{Re}\{\alpha\} = 0 \tag{2.13}$$

For this specific downconversion $\alpha \overset{!}{=} j\omega_c$, where ω_c is the carrier frequency in radians: $\omega_c = 2\pi f_c = 8.52 \cdot 10^7\ [rad/s]$. The mathematical procedure was done in MATLAB using state-space- (ss) and transfer-function-objects (tf), and computing the exact analytic result (not evaluated numerically) via symbolic toolbox[14]. The Laplace variable s is replaced via `subs()`-command to be $(s - j\omega_c)$ instead. After the shift, the new transfer function is evaluated in a relevant frequency range for spectrum display (see Figure 2.10). This figure shows – exemplarily for all other k – the shift to baseband *only*, which corresponds to “step 2” as explained in Section 2.2.1. The mirror copy has not been removed yet. As can be seen in the figure, the inherent discontinuity at frequency 0 of the passband system (blue) is still existent, just shifted to $-\omega_c$ (dashed black line). The shift operation on its own does not remove the symmetry of the original system, it rather displaces it by a constant offset ω_c. Since the original symmetry is always even around 0 in the magnitude spectrum (because the passband is real-valued), the shift will inevitably result in an odd spectral symmetry (red curve).

Symmetry also defines the position of the poles and zeros in the Laplace plane. The passband system (real-valued) can only have zeros and poles *on* the real axis or in pairs symmetrically around it, which is denoted *complex conjugation* (see Figure 2.11, blue and red symbols). After the shift, this is no longer the case. Rather, the relational axis of symmetry is displaced, so that it lies on a line parallel to the real (horizonzal) axis (see Figure 2.11, green line). The two poles – representing the two resonances around the carrier – and one significant zero (pattern in the gray box) are shifted to come to lie unsymmetrically around the x-axis after the shift (same pattern turned to green symbols).

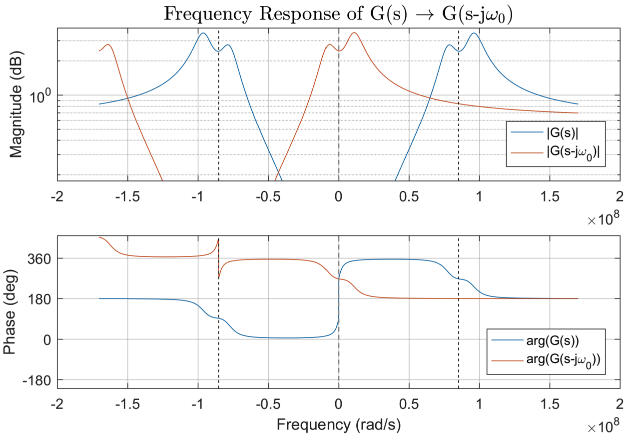


Figure 2.10: RFID transfer function model shift (without simplification yet); for k = 0.23

---

[14] MATLAB code for the downconversion of an exemplary RLC series resonance system is contained in Appendix A. The RFID system downconversion works in the same way (and results are very similar for $k \lll$), but cannot be included: system model data files are non-printable and proprietary.

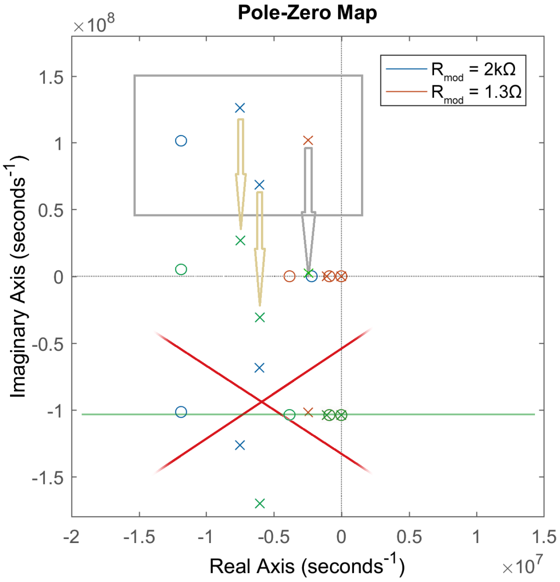


Figure 2.11: Downconversion by pole/zero shift and selection; $k = 0.52$ (poles can be clearly distinguished then). Two procedures in one plot: 1. "Mirror copy" removal before the shift (blue and red markers are fixed → then remove blue and red markers under the great red cross); 2. Shift occurs first (all green markers). Either way, the remaining green markers in the figure center (3 poles, 1 zero) are the final result.

The final step of downconversion is the removal of the "mirror copy". This can be done either *before* the shift (removing the blue and red poles and zeros in the area marked by the red cross) or *after* the shift (removing the green symbols on the green line and the ones shifted out of the plot – one green pole of the group is still visible at the very bottom of the plot). Only those poles and zeros are kept that come to lie close to the real axis after the shift (previously mentioned turned-green pattern), the two poles and one zero of the former blue system, and for the red system just one pole (compare gray box). If the selected poles and zeros are close enough to the real axis, the baseband behaviour of the system is narrowband, which means the remaining poles and zeros can be removed without affecting baseband model accuracy. However, selected poles moving away from the real axis as the overcoupling increases will be more and more affected by the outlying poles and zeros, and removing them will impair the baseband model. Figures 2.12 show this issue. Although the resonance frequencies are accurate and the phase is correct (except the irrelevant 360° offset), their magnitudes are out of proportion. The removal influenced the overall system gain and lifted the attenuation of the lower resonance.

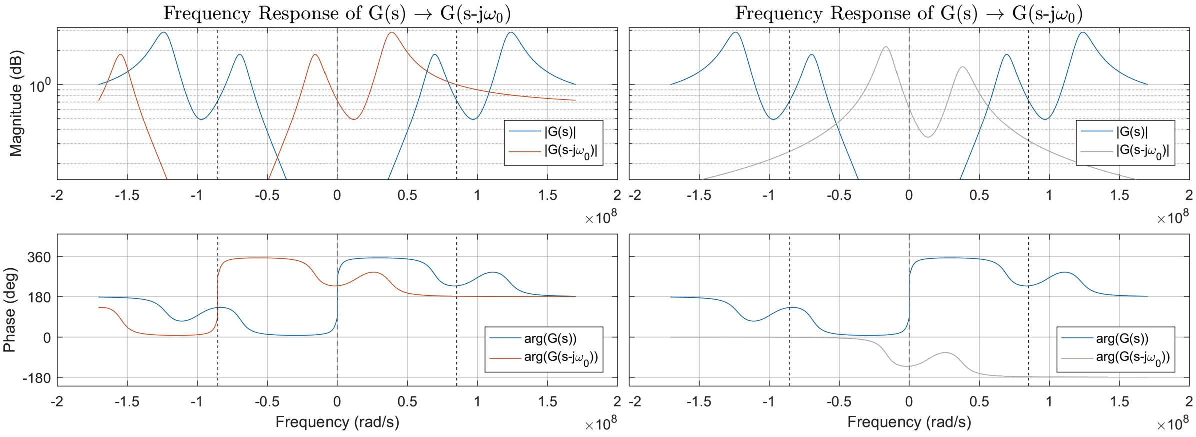


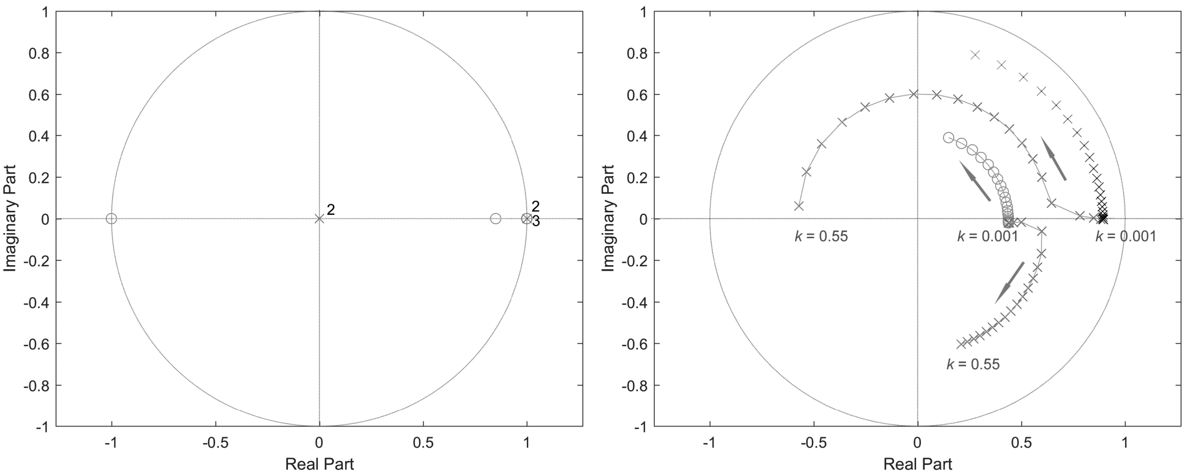
Figure 2.12: RFID transfer function model shift for $k = 0.52$; shift only (left); poles/zeros removed (right)

Modelling baseband behaviour of the system by pole and zero placement is thus a convenient approach, especially for lower values of k where the system is not overcoupled. As overcoupling increases, other issues arise that render the slight magnitude imbalances of Figure 2.12 insignificant: The received message signal will be *sampled*, so will be the system distortion in it. The upper resonance will likely move beyond the Nyquist frequency for $k > 0.5$, resulting in nastier effects than a small magnitude change at a well known frequency. So in summary, it does make sense to hold on to the Zero-Pole-Gain (ZPK) model (2.14). The continuous-time definition is stated here, a discrete-time variant is used in the final model (see Section 2.4.1).

$$G(s) = K \cdot \frac{\prod_{i=1}^{m}(s - z_i)}{\prod_{i=1}^{n}(s - p_i)} \qquad m \leq n \tag{2.14}$$

2.2.3 Discretization and zero/pole analysis

Both the shift-only and the reduced transfer functions are discretized for analysis purposes. The MATLAB function `c2d()` converts the continuous-time transfer function model (Laplace domain) to a discrete-time transfer function model (Z-domain) by sampling it at $f_s = f_c = 13.56\,\text{MHz}$ and using the matched pole-zero method for SISO systems. In case of the shift-only model, the discretization fails in the sense that the system cannot be used to pass a signal through in any meaningful way. It is nevertheless useful in order to show that the remaining outlying zeros and poles are in fact static – they do not change when k changes (see Figure 2.13a). They are thus not directly important regarding the events at baseband. They can be either removed or at least simplified. Keeping them is not very practical, because their position is so close to the unit circle that numerical stability cannot be guaranteed.



(a) Poles and zeros invariant to k

(b) Significant poles and zeros changing with k

Figure 2.13: Poles and zeros in z-plane

In case of the reduced transfer function, these static zeros and poles have been removed. The black (unconnected) sequence of poles in Figure 2.13b stems from the system with $R_{mod} = 1.3\Omega$. The two remaining variable poles represent the two system resonances shown before in left Figure 2.12 (gray, system with $R_{mod} = 2k\Omega$). There is one variable zero. Figure 2.13b shows the variation over all k, starting with $k = 0.001$ and proceding along the arrows. It is of great value for the analysis, because it conveys a very simple yet elegant way to represent the analyzed system using just a few parameters. This figure will be refered back to in Section 2.4.3 where the final system model will be evaluated.

2.2.4 Differential transfer function

At this point, the desired model of a compact $H_{baseband}(f)$ as described in the end of Section 2.2.1 is almost at hand. "Almost", because the conversion to baseband has been completed – however, there is not a *single* resulting system model, but in fact *two* separate models: one for each state of R_{mod}. Each model response – on its own – could theoretically be used as a $H_{baseband}(f)$ model. But by using merely one, the other is left out completely. Half the system's behaviour is lost. Their – in the right way – *united* response encourages further investigation.

Attempts were made to combine both systems (revisit Figures 2.4, 2.5 and 2.6) to get a single transfer function that can be used according to (2.10). The idea of a "differential transfer function" led to combining the two separate transfer functions by substracting one from the other in frequency domain: $G_1(z) - G_2(z)$. The substraction is supposed to represent the difference of both R_{mod} states, thus the switching behaviour. However, the result showed three resonances and the simulated output signals did not resemble the reference signals.

Different approaches of adding or substracting the only magnitude spectra proved futile. A simple or at least envisionable combination method of those two transfer functions did not manifest in the given time frame. Therefore, the attempt to directly combine the two transfer functions has been discontinued. An alternative approach is chosen, involving *system identification* methods and the given state space model. This approach does not have to deal with combination issues, because it identifies a single response for $H_{baseband}(f)$ in the first place.

The only way to properly identify the overall system behaviour is by evaluating the state space differential equations in time domain (i.e., using MATLAB function lsim). The input $u(t)$ of the state space model is the carrier source signal. In contrast, the message input – the *actual* input for the *desired* model – is incorporated in a rather nonlinear way: The two state space systems for both R_{mod} states are precomputed and hard-switched according to the binary message signal. The internal states $\boldsymbol{x}$ (voltages, currents) of the state space model are, however, exactly equal in both systems. So by continually tracking those internal states, then – before a switch of R_{mod} – *saving* them, and after the switch using them as initial states for the newly active system, the load modulation process and especially the system's distortion during this process are sufficiently accounted for.

This method is computationally expensive. It is impractical for later equalizer evaluation. However, system identification becomes viable, if the input is simply assumed to be the binary message switching signal. This involves a detachment from the classical state space theory. But since this binary signal can be *assigned* by choice, it is possible to test the system with known inputs. System outputs can be computed for these known inputs – the system can thus be identified: a system which is based on the "consummate" state space model (and thus exhibits properties and behaviour shown in the previous overall system analysis), but which is also directly fulfills the requirements for $H_{baseband}(f)$.

The input signals for system identification are constrained to be binary in this context (two amplitudes). Common choices of (multi-amplitude) input signals for system identification (e.g., sinusoidal sweeps, white noise) are thus not available – unless one is willing to pre-compute a sufficiently large number of state space systems for all values between $1.3\Omega < R_{mod} < 2k\Omega$, which is even more time-consuming. The means of choice are pseudo-random binary sequences (PRBS), special cases of which are "MLS" and "IRS".

2.3 MLS / IRS System Identification

MLS system identification is a method that uses pseudo-random binary sequences (PRBS) of a specialized kind: "maximum length sequences" (MLS). While PRBS are designed to mimic random stochastic behaviour in general, MLS try to adopt characteristics of white Gaussian noise specifically. A Gaussian noise signal is truely random in both amplitude and time. An MLS signal is completely deterministic and has two amplitude states only (± 1). However, with sufficient length L of the sequence a good approximation is possible.

2.3.1 Relation to Gaussian statistics

White Gaussian noise is widely used as an input to a system that is to be identified. Similarly to the direct measurement of the system's impulse response (via a short impulse $\delta(t)$), Gaussian noise shows a single peak – just like an impulse – in its autocorrelation function (ACF)[15]. The single peak is located at lag $\tau = 0$ of the ACF, and all other function values are zero (for the ideal Gaussian), because comparing the random Gaussian signal to time-shifted versions of itself (which is what the ACF does) will never show any similarity (correlation) – since it is random. The only exception is lag $\tau = 0$, in which case two unshifted versions are compared. The resulting value is equal to the noise variance $\sigma_n^2 = r_{nn}(0)$. Assuming the system is LTI (or almost LTI, at least during the measurement), the input noise (i) is correlated with the output signal (o), which contains the system influence (h):

$$r_{io}(\tau) = \Big(r_{ii} * h\Big)(\tau) \tag{2.15}$$

This equation simplifies even more for Gaussian noise input, because a convolution with $r_{ii}(\tau) = \sigma_i^2 \cdot \delta(\tau)$ merely scales the system response h by a factor σ^2. If deliberately set to $\sigma^2 = 1$, the desired system response *directly* results from the crosscorrelation: $r_{io}(\tau) = h(\tau)$. Otherwise, the input variance σ^2 can easily be measured and divided from h. The system has been identified.

2.3.2 Maximum length sequences

MLS resorts to the same procedure. The sequence is generated using linear-feedback shift registers. These registers are set up to repeatedly cycle through all states of the register of size L, yielding $N = 2^L - 1$ possibilities (which is the *maximum*, hence maximum length sequence; the all-zero state is excluded, because it breaks the cycle). The cycling is done so that: *a)* 1's and 0's occur equally often in the output sequence – since the number of possibilites is always odd, there is one more 1 exactly; *b)* occurence of consecutive patterns of 1's and 0's is unique (details see [27, p.43]). Consequently, the autocorrelation of such a sequence can be expected to be quite small for all time lags $\tau \neq 0$. However, the autocorrelation of MLS is not – as opposed to the Gaussian one – *zero* everywhere except for the peak at $\tau = 0$. If it were possible to create infinitely long registers, their autocorrelation would in fact be identical. But since they are finite, a constant offset term of $-1/N$ remains for all $\tau \neq 0$ (this is due to the excluded all-zero state). Since the MLS is a binary sequence, the autocorrelation is given in discrete-time form ($\tau \rightarrow\ l$):

$$r_{xx}[l] = \frac{1}{N}\sum_{n=0}^{N-1} x[n+l]x[n] = \begin{cases} 1 & \text{if } l = 0 \\ -\frac{1}{N} & \text{if } l \neq 0 \end{cases} \tag{2.16}$$

---

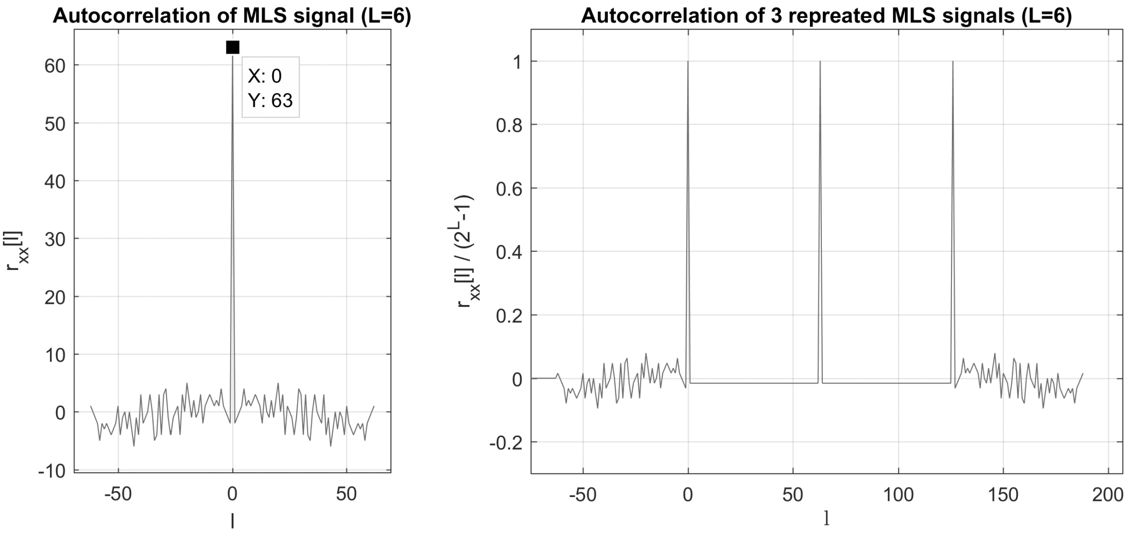
[15] Definitions ($\in \mathbb{C}$):
$r_{xx}(\tau) \;= \int_{-\infty}^{\infty} x(t+\tau)x^*(t)dt$ (continuous)
$r_{xx}[l] \;= \sum_{n\in\mathbb{N}} x[n+l]x^*[n] \approx \frac{1}{N}\sum_{n=0}^{N-1} x[n+l]x^*[n]$ (discrete)

It is evident that the longer the sequence, the closer will its ACF resemble the Gaussian one – hence becoming a Kronecker delta $\delta[n]$ (see Table 2.1). Once settled for an appropriate trade-off length L, (2.15) can be used for system identification, since it is equally valid in the discrete time case by changing (τ) to $[l]$.

$L \uparrow$	$N \uparrow\uparrow$	$r_{xx}[l] \longrightarrow \delta[l]$

Table 2.1: Parameter change for the MLS-ACF to approach Kronecker delta

There are some practical considerations that need to be taken into account first. When numerically computing the autocorrelation of a finite discrete signal such as the MLS, it is generically biased. The bias increases with l moving away from 0, the ACF becomes unreliable approaching the outer boundaries (see Figure 2.14) because of implicit zero-padding beyond those boundaries. A safe way to prevent the bias from influencing the system identification is to let the MLS cyclically repeat – not just once (as in Figure 2.14), but at least 3 times in a row.[16] Then, the middle spike is free from the boundary effect, because the adjacent sequences pose as natural extensions. Only the constant offset term $-1/N$ remains, it can be seen in Figure 2.14b, where the constant line below the middle peak is slightly lower than 0. The low MLS order of $L = 6$ has been chosen for these example figures to clearly show the offset and the boundary effects.



(a) ACF of 1 MLS, raw & unscaled

(b) ACF of 3 repeated MLS, scaled

Figure 2.14: Discrete autocorrelations (MLS)

The identified system impulse response will be located around the middle peak (desired location), as well as around the neighbouring peaks (undesired repetitions). In order to get clear results, the length of the MLS must be chosen such that the tails of the system response decay within the given frame of samples (in the example case of $L = 6$, the maximal tail length may be $N = 2^L - 1 = 63$ samples[17]). Otherwise overlap occurs and irreversibly damages the measurment. This overlap is called “time-alisaing”. It is recommended in [28] that L be chosen to at least accomodate double of the expected tail length, because the measurement resolution increases and so does the SNR. The SNR can additionally be increased by repeating the MLS more than 3 times (those repetitions will be as clean as the middle peak in Figure 2.14b) and

---

[16] this cyclic extension may be expressed by the modulo operator: $x_{MLS}[n \mod N]$; also relevant for (2.17)

[17] for causal systems

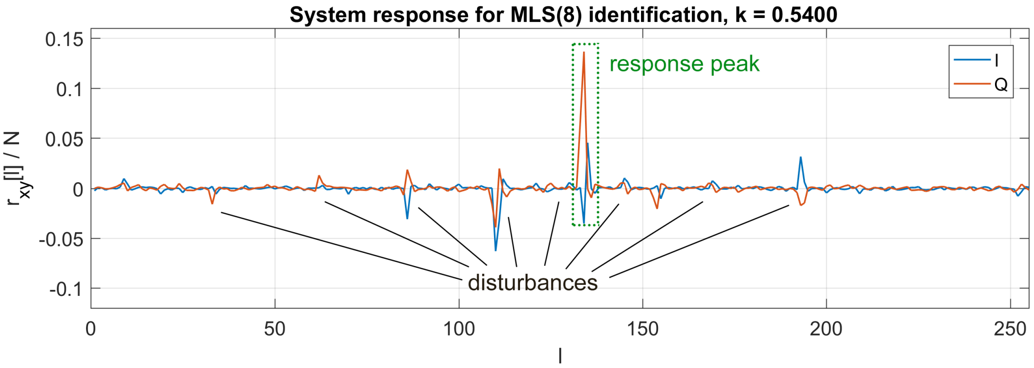


Figure 2.15: MLS system identification issue: disturbance peaks

averaging over multiple clear measurements. Additive noise is reduced by averaging. Besides that, MLS is inherently more robust to noise compared to other measurement signals in this context because of its optimal crest factor / peak-to-average power ratio (PAPR)[18] of 1.

The MLS system identification originates in the 1960's [29], since then it has been applied in many fields, one of them being audio: Borish [30] and later Vanderkooy [31, 32] explored the idea of identifying acoustic impulse responses. On their journey of research they came across "spiky distortions" within the correlation function of MLS sequence with the output. Dunn and Hawksford [33] measure and study the distortion, and trace it back to nonlinearities. Wright [34] analyzes MLS reaction to nonlinearities by invoking Volterra series. It is proven there that MLS system identification is extremely susceptible to even weak nonlinearities in the measured system.

The higher-order Volterra kernels are having a similar effect on the raw binary sequence as the ACF: they contain multiplied versions of the sequence with itself. With increasing order, the Volterra system exhibits nonlinear multi-dimensional convolution, which – like the ACF – implies shifting the sequence over itself. Once passed through a Volterra system, the MLS is thus superimposed by differently weighted and most importantly time-shifted versions of itself. If the cross-correlation between the original input and the nonlinear output is now computed according to (2.15), the original correlates multiple times because of those superimposed versions. The result is that several other peaks appear in the crosscorrelation besides the desired one (see Figure 2.15), which are distributed arbitrarily across the range depending on the nonlinearity. However, their positions are deterministic, which means those peaks are reproducible, thus their location may be anticipated to some extent. On the downside, they cannot be averaged out due to their deterministic location.

Figure 2.15 shows the MLS measurement of the RFID system with high coupling. The disturbance peaks may even be higher than the one containing the actual system response. But most significantly, the smaller disturbances do influence the area of the desirable peak, rendering the measurement useless. Figure 2.15 shows the extreme case in the RFID system context. For lower coupling factors k, the amplitudes of the disturbing peaks decrease – but not gradually; rather erratic (reduce in one location, but grow briefly in another one). Thus, the impact of nonlinear disturbance depends on how strong the input amplitude is. Unfortunately, this amplitude is not adjustable in the RFID system: The resistor is switched between its two states – and the output amplitude ultimately depends on the coupling k. These circumstances call for a refined method able to suppress nonlinearities.

---

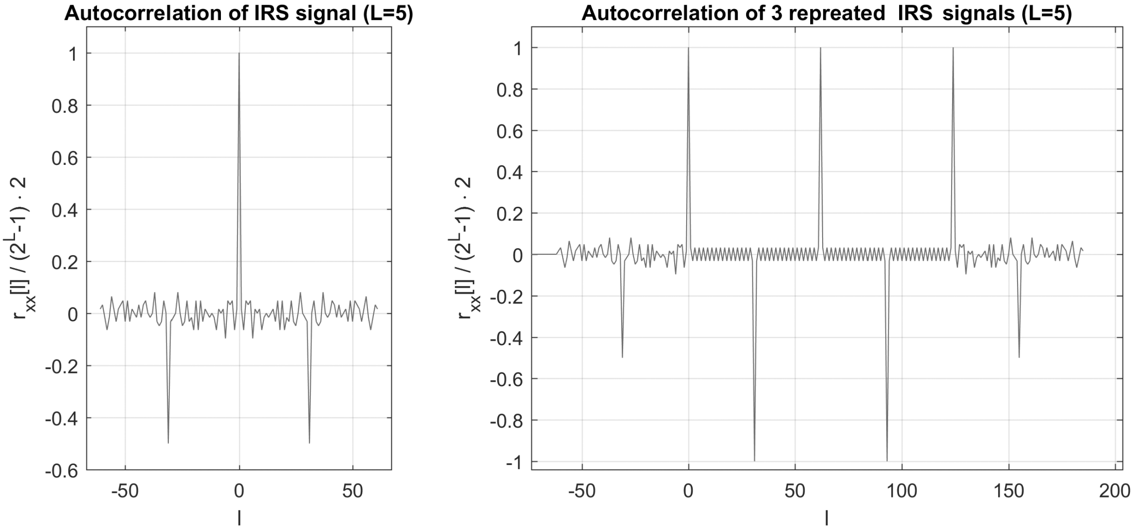
[18] Crest factor $C = \frac{|x_{peak}|}{x_{rms}}$; PAPR $= \frac{|x_{peak}|^2}{x_{rms}{}^2} = C^2$

2.3.3 Inverse repeat sequence

The simple and straight-forward solution is called **inverse repeat sequence** (IRS). It is first mentioned in [35]. According to Stan [28] the new sequence that replaces MLS as the input signal is actually re-composed of MLS sequences:

$$x_{IRS}[n] = \begin{cases} x_{MLS}[n] & \text{if } n \text{ is even, and } 0 \leq n \leq 2 \cdot N \\ -x_{MLS}[n] & \text{if } n \text{ is odd, and } \ 0 \leq n \leq 2 \cdot N \end{cases} \tag{2.17}$$

The IRS sequence has double the period length $(2 \cdot N)$ of the MLS signal of order L. To clarify the equation, x_{IRS} is generated by first concatenating two periods of the MLS signal so that their overall length is $2N = 2 \cdot (2^L - 1)$. In a second step, the sign of every second sample is inverted. This is important to note here, because (2.17) assumes x_{MLS} to be periodic in N without explicitly mentioning it (see also [16]).



(a) ACF of 1 IRS, scaled

(b) ACF of 3 repeated IRS, scaled

Figure 2.16: Discrete autocorrelations (IRS)

The ACF of IRS is shown in figure 2.16. The order $L = 5$ is chosen so that the length of the resulting IRS is equal to that of the MLS shown previously in Figure 2.14. Since the underlying IRS is built of two MLS periods, the ACF shows two additional (besides the one at $l = 0$) correlation peaks – which are negative because of the alternating sign: For the two sub-periods of $L = 5$, $N = 31$. According to (2.17) all odd samples n are inverted, so the ACF shows negative peaks at $l = \pm 31$. The resulting IRS period is $2N$, which will always be an even value, so the main peak at $l = 0$ is positive.

Far more significant is the content of Figure 2.16b. Due to the alternating sign, the formerly constant offset of $1/N$ transforms into an oscillation at Nyquist frequency $(f_s/2)$. The IRS has even length of $2N$ and – because of the sign change – both signal amplitudes ± 1 now occur equally often. This means the IRS as well as its ACF are zero-mean, thus far less susceptible to nonlinearities in the measured system. It has been shown in [33], that the IRS completely rejects any effects caused by even-order nonlinearities (polynomial powers of 2, 4, 6, etc.), only odd-order terms have an influence (equal influence as in case of MLS). This property gives IRS a significant edge over MLS. And it comes merely at the cost of needing a double length input signal and thus a longer crosscorrelation. Methods that applied to MLS (averaging out noise,

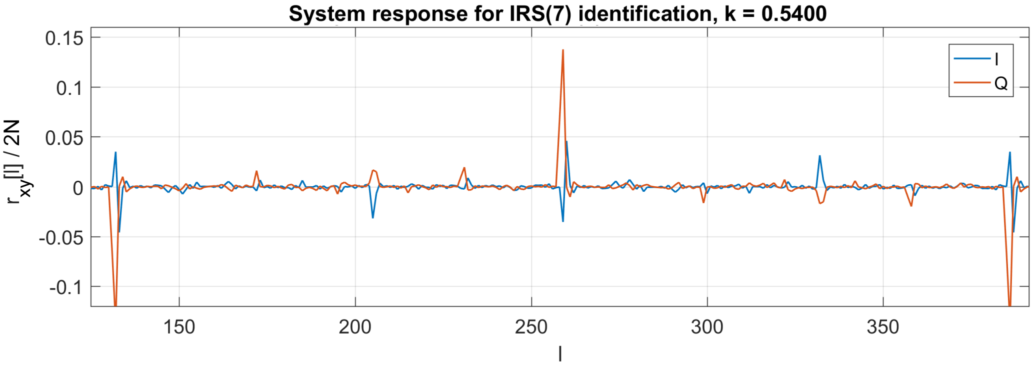


Figure 2.17: IRS system identification issue: disturbance peaks

cutting out the relevant area around the main peak) also apply to IRS. Just for completeness, there exists a third alternative using a ternary sequence, which additionally rejects third-order nonlinearities. This alternative is, however, not applicable here, since the R_{mod} can only assume two states.

Both MLS and IRS have been implemented in order to identify the repsonse of the given RFID system. Sequences of orders 6 - 11 are pre-generated and stored in MATLAB data files, containing the underlying original binary sequence, a sequence repeating the original up to a limit[19], and this same long bit sequence represented (more compact) as a hexadecimal string. During simulation (as previously described in Section 2.2.4), this hexadecimal string is translated into the signal controlling the R_{mod} switch. The system's output signal is then sampled at $f_s = 13.56\,\text{MHz}$ as a complex-valued signal (I and Q component). The (complex-valued) cross-correlation is computed according to (2.15) of this output signal and the underyling original sequence (real-valued). The exemplary MLS result has been shown in Figure 2.15, a directly comparable IRS result is shown in Figure 2.17. To the far left and right, the negative peaks can be observed (comp. Figure 2.16); they also contain the system response, but inverted. The distortion peaks here are located at similar positions (comp. Figure 2.15) relative to the main peak. In contrast to the MLS measurement, those distortions are far less severe in close vicinity to the main peak. Additionally, their position is more stable and thus predictable. Reliable system identification can be achieved by IRS. The main peak's vicinity becomes even cleaner with increasing IRS order. For the final measurement an IRS order of 10 has been chosen.

2.3.4 Postprocessing

A few postprocessing steps are necessary to isolate the impulse response. The peak is easily identifiable, because it has a deterministic position in the crosscorrelation. In order to automatically check if the position is correct, a **peak finding algorithm** is applied. It finds the index l of the peak based on the maximum of the absolute impulse response (I and Q combined) and uses a threshold relative to that maximum (to account for neighbouring high values).

As a second step the range of relevant samples after the peak has to be defined. Any sample beyond that range is set to zero, so that only the clean impulse response remains. Since the RFID system is composed of antenna resonances, the impulse response is theoretically infinitely long (IIR). It is decaying, however, at a reasonable rate, so the decaying tail of the response

---

[19] The simulation framework constrains the length of the input signal (the message) over the whole communication process to about 9000 bits. The MLS order 11 ($2^{11} - 1 = 2047$ bits) is the highest possible order for which the sequence can still be repeated at least three times within this limit.

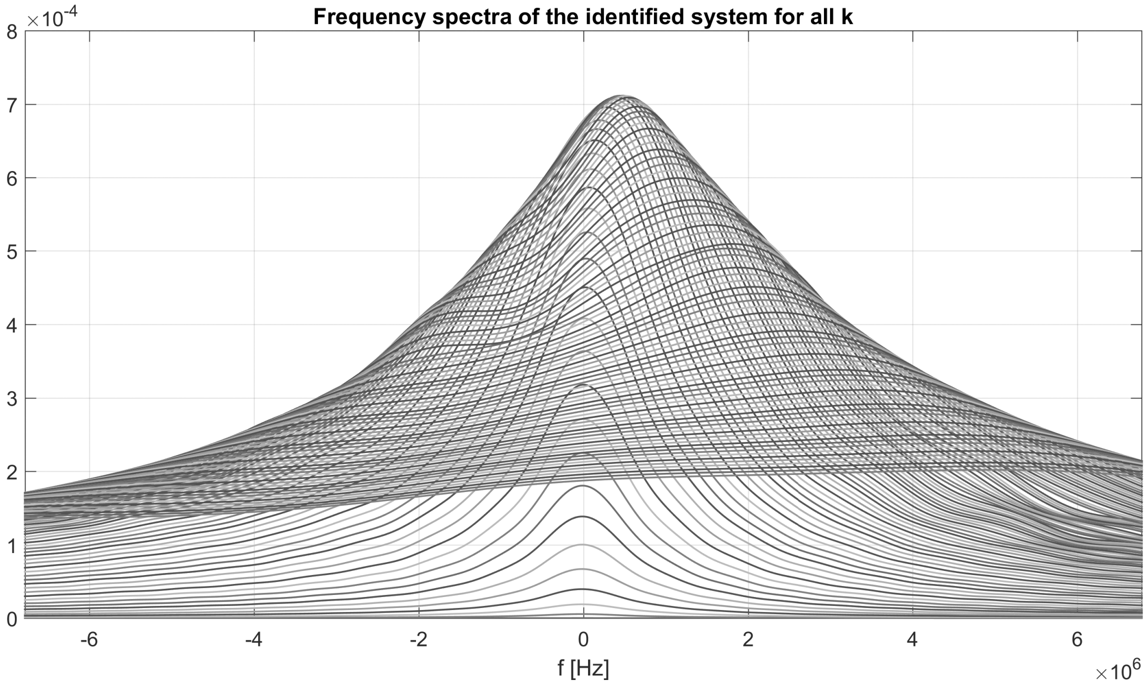


Figure 2.18: IRS system identification: Frequency spectra

may be truncated at some point. Examinations showed that the slowest decay occurs at low coupling factors $k \lll$. For $k = 0.001$ the decay can be assumed to encompass at least 40 samples before truncation is recommended. For increasing k the decay grows gradually shorter until it spans just $6 - 7$ samples at $k = 0.55$. Based on these observations, the length (in samples) of the **selection window** is interpolated for any k between the two mentioned bounds (MATLAB polyfit and polyval). The value of k is known during each simulation case; so it becomes possible to automatically cut the impulse response free. Optionally, the rear of the selection window can have a faded weighting: If hard rectangular truncation is not wanted, the fading window (second half of a Bowman window) smoothly blends the decay with the following zeros.

Averaging has been implemented as well. Instead of a single response peak, multiple repeating peaks are taken into account, which improves SNR in the presence of random noise in the measurement. But why add noise to the simulation in the first place? That would counteract a clear system identification. So all additional noise sources are disabled for the measurement and averaging is thus not necessary. For the noise caused by the nonlinearities in the system it has been mentioned in the previous sections that it is of deterministic nature. Averaging will not cancel these disturbances.

The results of the system identification are shown not as individual impulse responses in time domain, but rather their corresponding amplitude spectra in frequency domain. These spectra are computed from the postprocessed clean impulse response, via FFT. It is important to note that *both* the impulse response *and* the (overall) FFT result are generally complex-valued. No symmetry may be expected.

Figure 2.18 shows frequency responses for 100 cases of k linearly spaced between 0.001 and 0.55. Starting from the lowest k, the lines are plotted sequentially and overlay at some point. The lowest k curves are the ones at the very bottom of the figure. A single resonant shape can be observed at $f = 0$, so small that its magnitude is barely noticable by eye. With increasing k

this magnitude grows, the resonant shape evolves upwards in the plot, still centered at $f = 0$. Around an optimal $k \approx 0.15$ the single shape reaches its maximum and becomes maximally wide. Here the overlay starts. Amplitudes decrease again, and at the same time the shape splits into two resonances drifting apart from $f = 0$ towards higher and lower frequencies. The upper resonance shows a greater magnitude compared to the lower resonance. This relation remains valid; however, for k approaching 0.55 the response spans a wide frequency range – and the resonances are hardly perceptible as distinct resonances. The spectrum becomes more and more flat. If zooming in on the y-axis, the upper resonance peak for $k = 0.55$ lies at the Nyquist frequency of 6.78 MHz (right plot edge). This is in accordance to previous findings in this thesis (see Figures 2.4, 2.5 and 2.6). The data of the presented figure is sampled at $2f_s$, otherwise the curves for high k would already mirror back into the negative frequncies. The slight ripple on the right side ($f \approx 6$ MHz and amplitudes $1.5 \cdot 10^{-4}$) are caused by a small nonlinearity peak located within the still long decay tail at these particular k-values. The effect is minor. It can be overcome in the modelling process described in the following section.

2.4 Discrete-Time Zero-Pole-Gain Model

At this point an extensive data set is available. Impulse responses have been sampled for a 100 different k in the range of 0.001 and 0.55. Using these impulse responses as a system model directly can be cumbersome for several reasons: Loading and managing the data is unwieldy and takes up memory and processor resources; and furthermore the coupling steps k are fixed to these 100 cases. For k is the most important parameter, it is sensible to keep it *continuous*, that is, fill the gaps in between these 100 cases. By cycling through the range of k, a relationship is established between consecutive impulse responses. “Neighboring” responses are very similar to each other due to the already dense grid of k. Impulse responses in between k-values may thus be reconstructed by comparing the corresponding neighbors and then performing interpolation. Also, by taking this relationship into account, the nonlinear irregulatities (mentioned in the previous section) can be compensated for since changes in neighboring responses are expected to be smooth.

2.4.1 Model definition

A parametric model is chosen to represent the RFID system. Parametric models attempt to reduce the complexity of evaluating a system by choosing a sufficient set of parameters which properly represent the system in the given context and – most importantly – are able to accurately predict how the original system would react in that situation, and thus act as a simplified substitute. Based on the results of the system analysis, the chosen parameters are:

- $\mathbf{k}$: coupling coefficient, $k \in [0.001\ 0.55]$, $k \in \mathbb{R}$
- $\mathbf{p_1}$: pole of the lower antenna resonance, $p_1 \in \mathbb{C}$
- $\mathbf{p_2}$: pole of the upper antenna resonance, $p_2 \in \mathbb{C}$
- $\mathbf{z_1}$: zero signifying magnitude relation of p_1 and p_2, $z_1 \in \mathbb{C}$
- $\mathbf{z_2}$: zero needed for model estimate, outside unit circle, $z_2 \in \mathbb{C}$
- $\mathbf{g}$: model gain, $g \in \mathbb{C}$

This neat set of 6 parameters is assumed to sufficiently model the system’s behaviour. The model is **linear** and currently time-invariant, but may be extended easily at this point to a quasi-time-varying scope: Smoothly increasing k can be interpreted as moving the card or tag *towards* the reader, and decreasing vice versa.

The chosen parameters are inherent to a (discrete) zero-pole-gain model, abbreviated as ZPK model. The continuous ZPK model in Section 2.2.2 (see (2.14) on page 26) has already been found to be a convenient type of model for this thesis. It is defined in discrete-time as a special form of a rational transfer function. Usually, transfer functions are represented by numerator and denominator polynomials (z-domain):

$$\hat{H}(z) = \frac{\hat{B}(z)}{\hat{A}(z)} = \frac{\sum_{j=0}^{n_b} b_j z^{-j}}{\sum_{j=0}^{n_a} a_j z^{-j}} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \ldots + b_{n_b} z^{-n_b}}{a_0 + a_1 z^{-1} + a_2 z^{-2} + \ldots + a_{n_a} z^{-n_a}} \tag{2.18}$$

The ZPK form is the factorization of this equation. Both polynomials $\hat{A}(z)$ and $\hat{B}(z)$ factor into terms containing the roots of these polynomials – called poles and zeros respectively. Since all coefficients a_j and $b_j \in \mathbb{C}$, all factored root terms are in fact *linear* (no square terms as in conjugate roots). The gain factor $\mathbf{g}$ contains b_0 and a_0 so that the highest-powered z-term equals 1 before factorization.

$$\hat{H}(z) = \frac{\prod_{j=1}^{n_b} (z - \mathbf{z}_j)}{\prod_{j=1}^{n_a} (z - \mathbf{p}_j)} \cdot \mathbf{g} = \frac{(z - \mathbf{z_1}) \cdot (z - \mathbf{z_2})}{(z - \mathbf{p_1}) \cdot (z - \mathbf{p_2})} \cdot \mathbf{g} \tag{2.19}$$

The two representations are equivalent, they can easily be converted back and forth. The ZPK form has the advantage that stability can be verified directly ($|\mathbf{p}_j|, |\mathbf{z}_j| < 1 \quad \forall j$). Poles and zeros as the model parameters are prefered over the coefficients in $\hat{A}(z)$ and $\hat{B}(z)$, because they are more stable numerically: a_j and b_j are each a *product* of all poles, respectively zeros (binomial expansion). In case of the chosen polynomial orders here – that is $n_a = n_b = 2$ – the effect is not yet severe, but with longer polynomials it will be. Coefficients will thus have a far greater value dynamic (depending on the gain as well), while the zeros and poles are mostly located inside the unit circle (gain factor independent), ranging between $0 < |\mathbf{p}_j| < 1 \quad \forall j$ and $0 < |\mathbf{z}_j| < 1$ for most j (see ensuing Figures 2.19 and 2.20 on page 37).

2.4.2 Estimation algorithms

Common estimation algorithms reduce the problem to solving denominator polynomials only ("all-pole system") because it becomes a linear problem then. Additionally, symmetry and thus real coefficients are assumed to constrain the problem. From the filter design perspective, various methods exist to estimate numerator polynomials only ("all-zero system" or FIR). Given the following requirements

- estimate both polynomials $\hat{A}(z)$ and $\hat{B}(z)$ (equation 2.18)
- estimate them simultaneously
- assume all coefficients to be complex: $a_j,\ b_j \in \mathbb{C}$
- assume no symmetry = no conjugate roots in the estimated polynomials
- estimate based on impulse response data (or its corresponding FFT),

the list of available methods is quite short. MATLAB offers abstract function layers in the *DSP Toolbox* called `fdesign.arbmagnphase` (arbitrary magnitude and phase design), however, the underlying methods are not visible to the user. Visible functions supporting complex coefficients are `invfreqz`, `stmcb` and `prony`, which are documented and explained in [36]. The straight-forward approach would be to minimize the error when comparing the outputs of reference and estimate (unit input):

$$e_{out}[n] = y[n] - \hat{y}[n] \quad \Rightarrow \quad E_{out}(e^{j\omega}) = H(e^{j\omega}) - \hat{H}(e^{j\omega}) = H(e^{j\omega}) - \frac{\hat{B}(e^{j\omega})}{\hat{A}(e^{j\omega})} \tag{2.20}$$

The corresponding difference equation is nonlinear and thus difficult to solve directly. Prony's method linearizes the problem by multiplying the frequency domain equation in (2.20) by $\hat{A}(e^{j\omega})$ and solving without the weighting factor term. The error is then called *equation error* $E_{eq}(e^{j\omega})$ (due to its difference equation):

$$\hat{A}_{(weight)}(e^{j\omega}) \cdot E_{out}(e^{j\omega}) = \hat{A}(e^{j\omega})H(e^{j\omega}) - \hat{B}(e^{j\omega}) \tag{2.21}$$

$$E_{eq}(e^{j\omega}) = \hat{A}(e^{j\omega})H(e^{j\omega}) - \hat{B}(e^{j\omega}) \tag{2.22}$$

Prony's method (`prony`) first solves for $\hat{A}(e^{j\omega})$ *only* (neglecting the $-\hat{B}(e^{j\omega})$ term in (2.22)), and then finds $\hat{B}(e^{j\omega})$ by minimizing the output error under L_2-norm containing the now known $\hat{A}(e^{j\omega})$. The missing weighting factor has the effect that the error is spectrally weighted. The estimate is less reliable in frequency regions where the influence of the poles is prominent.

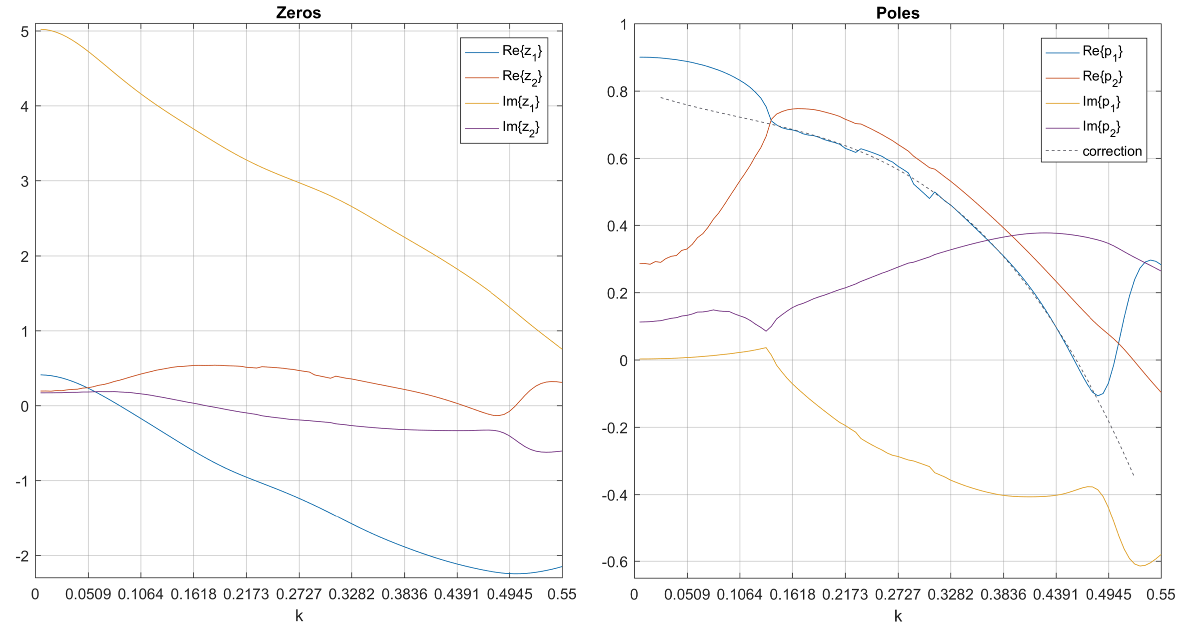
The Steiglitz-McBride method (`stmcb`) minimizes the output error equation (2.21) iteratively. It uses the estimate of the Prony method as a starting point. In each following step, the new estimate of $\hat{A}(e^{j\omega})$ is used as new “countering” weight function ($\hat{A}^{-1}_{(weight)}(e^{j\omega})$). It thus converges towards a minimized $E_{out}(e^{j\omega})$. Both methods solve in time domain (using impulse response and difference equation).

As opposed to these, the algorithm in MATLAB's `invfreqz` solves the minimization of the equation error in frequency domain. It is faster (because it uses FFT), but numerically less accurate. With increasing FFT length, however, the result approaches the Steiglitz-McBride solution. `invfreqz` needs an explicit declaration for the coefficients to be complex. A MATLAB code example is given in Appendix B comparing the three methods. The model parameters have been estimated using `invfreqz`.

2.4.3 Model adjustment and validation

When plotting the estimated model parameters each against k separately (see Figure 2.19), it becomes clear that they are meant to change in a continuous fashion – which makes sense considering the underlying continuous-time RFID system. However, there are regions of deviation. These can be identified as errors from the new point of view, and can be corrected.

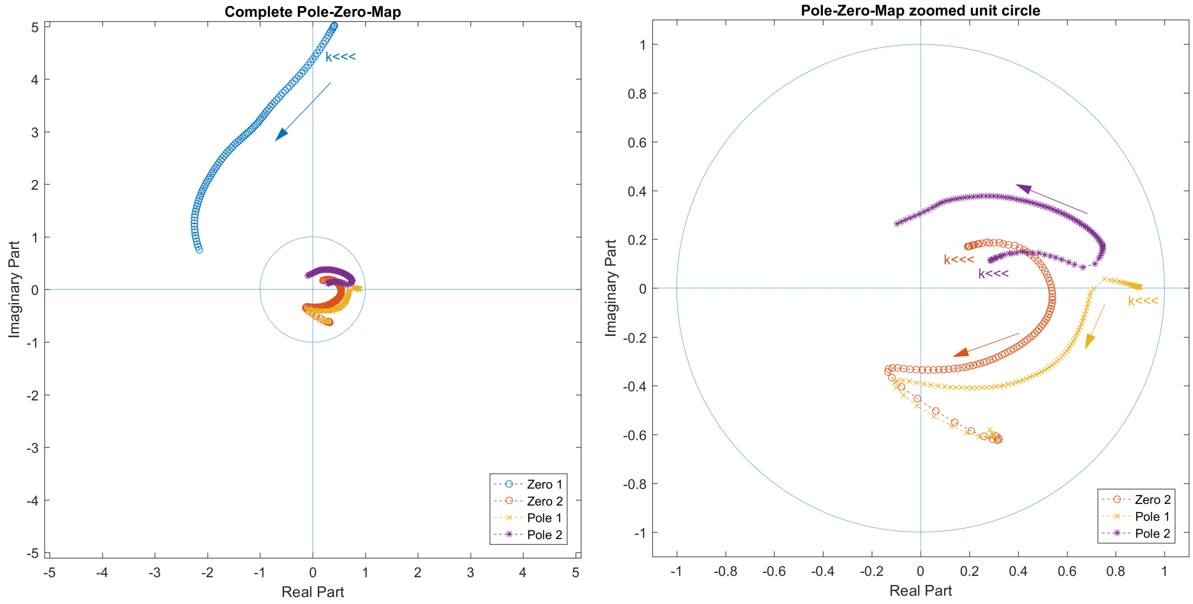
Figure 2.19 shows the values of all model zeros and poles, split into real and imaginary parts, for all k in the considered range. At very low k the fluctuation in some pole values are due to numerical limits: The sampled difference between the R_{mod} states is close to floating point underflow, where fine resolution is not possible. This error propagates through all steps up to this point, where it can be corrected. At $k \approx 0.14$ all poles show an abrupt change. This change marks the point of *critical coupling* in the system, it is significant and must be retained. In between $0.20 < k < 0.35$, errors occur (best visible in the real part of pole p_1, but existent in all roots) which can be traced back to the nonlinearities discussed in Section 2.3.4 and visualized in Figure 2.18. This major error region ($0.20 < k < 0.35$) is corrected by appropriate methods (mostly fitting of polynomials of orders 3 and 5). One example of such a correction curve is shown in Figure 2.19b for $Re\{p_1\}$. At $k > 0.5$ the system's spectrum is so wide that the estimation algorithm (`invfreqz`, see Section 2.4.2) reaches its limits in terms of accuracy. Because spectral magnitude variations become more subtle with increasing k, poles and zeros (transfer function coefficients respectively) are harder to estimate by the chosen spectral method.



(a) Zeros

(b) Poles

Figure 2.19: Poles and zeros: Uncorrected variation over k



(a) Full Z-plane view

(b) Zoomed to unit circle

Figure 2.20: Poles and zeros: Locations in Z-plane for all k

The final step is to detach the model from the discrete data points (100 values of k), of course without losing validity, but with the objective of having a more convenient model representation to work with. To this end, the poles and zeros (curves shown in Figure 2.19) are replaced by smoothing splines. This is an interpolation method that keeps all original data points of k as “knots” and that uses piecewise polynomials to produce a smooth, continuous alignment for the resulting curve(s). The spline representation reduces the “size” of the model and is able to produce parameters for *arbitrary* k in the range of 0.001 to 0.55 (and possibly even a little beyond that). The smoothing splines also compensate for the aforementioned inaccuracies at low k.

The poles and zeros of the final model are shown in Figure 2.20 in the complex plane. The starting point for low k are marked as $k \lll$, the arrow points towards increasing k values. Especially the unit-circle-subfigure (Figure 2.20b) bears remarkable resemblance to the previous Figure 2.13 on page 26 in Section 2.2.3, which showed the poles and zeros of the baseband-shifted state space model. A comparison to Figure 2.20b reveals the dominant resonance pole starting close to $Re = 1$ (on the real axis, which signifies frequency $f = 0$; start of yellow stretch), and the other (inferior, violet) pole which – considering Figure 2.13 – should also start in the real axis. It evidently starts only in the vicinity of the real axis; however, being that close to the (red) zero, its influence on the model transfer function is entirely negligible (the yellow pole determines the transfer function for low k), so its exact position is not important for the resulting model response (for low k). For increasing k, the pole successions are similar in the two compared figures, until the yellow pole has a shift in direction. At this point, the pole should proceed on its approximately circular course towards the negative real axis. It does not so because of the previously mentioned effect of a widened spectrum on the estimation (invfreqz). Fortunately, the (red) zero cancels the yellow pole at high k, so the impact on the transfer function is negligible in this case also. As a concluding remark on the comparison let it be said that these similarities only *indicate* model validity. They are no proof of it, because the compared Figure 2.13 merely shows the poles and zeros for the *separated* states of R_{mod}, whereas the conceived model consists of a combination of both states of R_{mod}. In the end however, the model is very plausible, because of the conclusions drawn from the system analysis: The state $R_{mod} = 2k\Omega$ does produce a stronger resonant effect in the card's circuit (see Section 2.1.2), thus will determine the *combined* system's response in a stronger way than the state $R_{mod} = 1.3\Omega$. From the results presented here, it is clear that both approaches to the $H_{baseband}$ model are closely related; however, attributes or defining characteristics (mathmatically) of this relationship cannot be deduced easily. The operation of combination of the two states as examined in Section 2.2.4 thus cannot be made up for. Findings point to a nonlinear way of combination that may also depend on k.

The output waveforms of the accurate state space simulation and the conceived model are compared in Figures 2.21, 2.22 and 2.23 in a second validation effort. The input is an IRS sequence, same for both systems. The state space response is simulated in (continuous) time-domain (lsim) and sampled at $f_s = 13.56\,\text{MHz}$. The model response is already equivalent to a sampled baseband response, and is provided by a dedicated MATLAB function called `[B,A,h,ZPKm] = AF_generateSplineModel(kc)`, where `kc` is the chosen value of k and the returned variables are either the ZPK model parameter structure (`ZPKm`), the transfer function coefficients (`B,A`) or the (truncated, FIR) impulse response `h`. Both `[B,A]` and `h` can be used to filter (or convolve with, respectively) the given input IRS sequence. The coefficients (`B,A`, IIR response) were used in the plots.

Figure 2.21 represents all cases below critical coupling, for which the model is very accurate. Above critical coupling the nonlinear effects in the state space simulation start to show, however, strong similarities are still evident in Figure 2.22. Finally, in Figure 2.23 the nonlinearities manifest themselves in spiky transient overshoots and a tendency to a clipping-like behaviour in the state space simulation. The model, on the other side, is "sweet-tempered" and shows an output that much resembles the IRS input sequence (since model impulse response is very short at this k). Another detail to notice here is the imaginary part (red), which now has a greater amplitude than the real part of the signal (blue). The system thus approaches a 90° phase shift with increasing k.

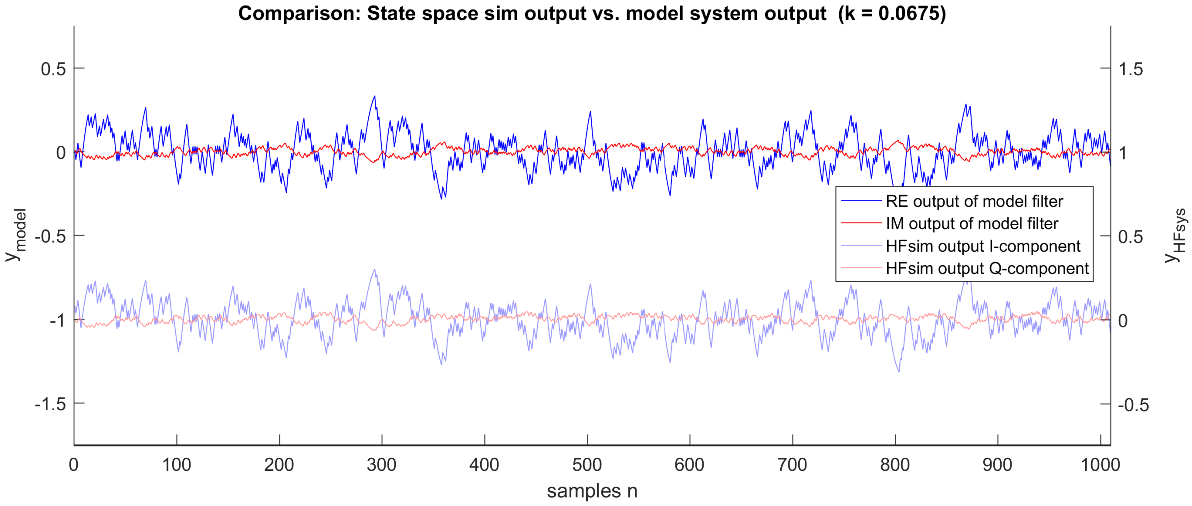


Figure 2.21: Model validation by output waveform comparison for $k = 0.0675$

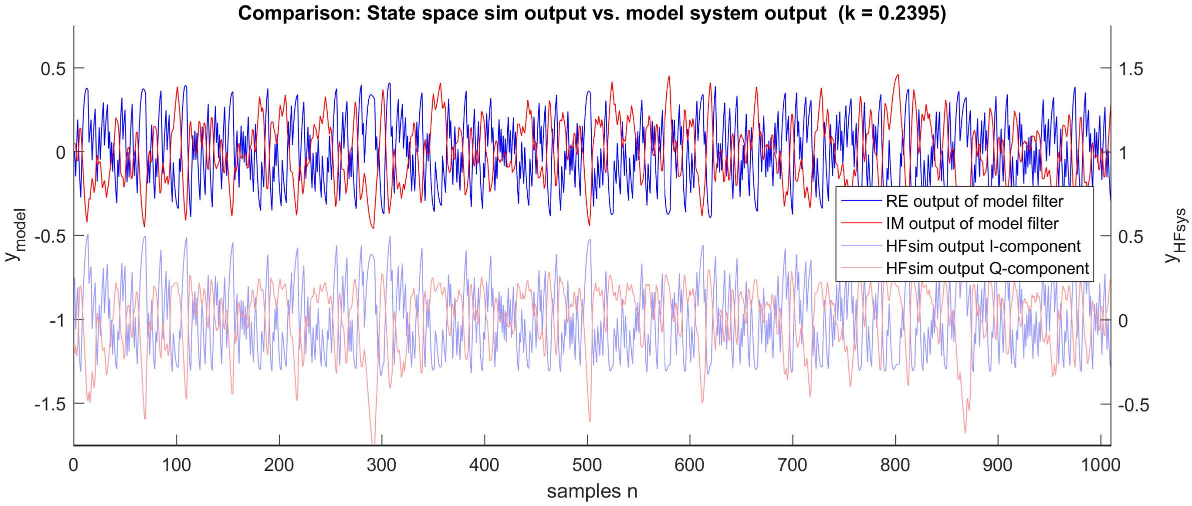


Figure 2.22: Model validation by output waveform comparison for $k = 0.2395$

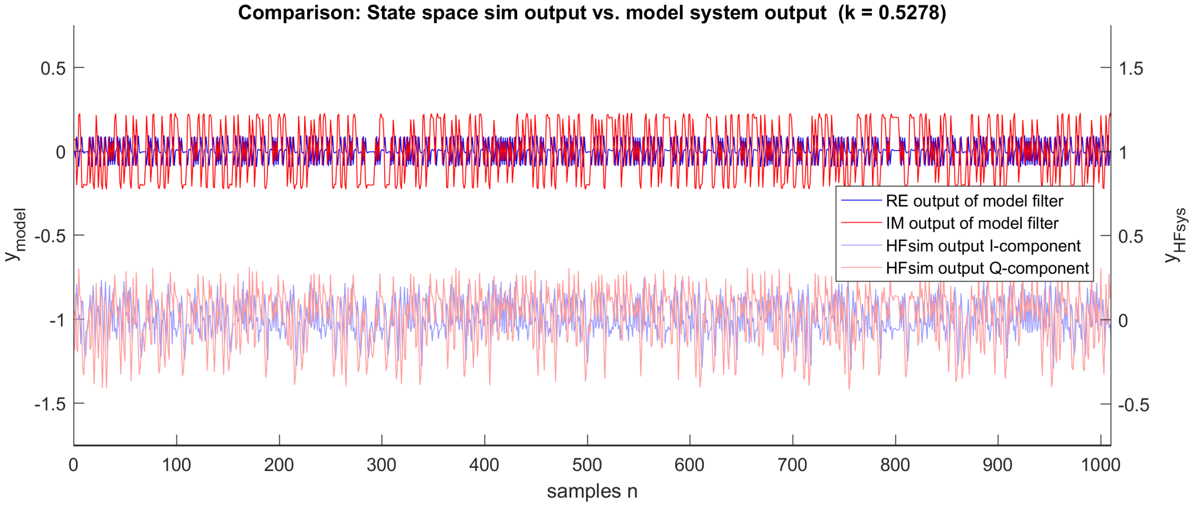


Figure 2.23: Model validation by output waveform comparison for $k = 0.5278$

2.4.4 Concluding notes on the model

The modeled system is stable and causal. One zero (blue stretch in Figure 2.20a) is located outside the unit circle, making the modeled system *non-minimumphase.* The exact term for this kind of system is “mixed-phase”, because it has zeros both inside and outside the unit circle. This property is significant when it comes to finding an inverse system – a system that can undo the effect of the given RFID system channel ⇒ an equalizer – because *non-minimumphase* systems have an inverse which is unstable (but at least causal). As a major consequence, any equalizer inserted after such a system will always *approximate* the inverse.

As explained in Section 2.4.1, the model is linear. It only represents the linear part of the system. The nonlinear portions are not accounted for. However, equalizing only the linear part is hypothesized to achieve great improvements in message reconstruction (see Section 3.3.2). The model validity is very high for lower k and decreases slightly as k approaches 0.55. Linear equalization is far more efficient in terms of implementation and computational cost compared to nonlinear equalization. In the given case, the magnitude of impact of nonlinear effects does not justify these higher costs of a model with higher complexity. Further potential model refinements concerning the nonlinear effects are thus discussed in the thesis’ outlook (see Section 5.2).

3

Equalization

In the context of communications, the term *equalization* refers to a process which offsets all negative effects caused by a given transmission channel. This opening statement is rather general. The desire to undo such negative effects is encountered in basically every application in the field of communications. Equalizers appear in many different fields as well – for example in *audio engineering*, where they aid in restoring or enhancing certain features in sound signals. The general scheme of equalization can thus be applied in many contexts. The difference in each case lies in what is considered the "negative effect".

In this thesis, this "negative effect" has a name: *intersymbol interference*. It has been described briefly in the introduction (see Section 1.1.3), visualized in Figure 1.3. This figure shows the effect of the given RFID channel (small coupling k) for a single pulse of the message signal. The pulse is not confined to its own "time slot" after it has passed through the system. Instead, it is smeared across several neighboring time slots. Although the peak amplitude is still close to the original time slot, a significant portion of this pulse's energy influences subsequent pulses. Neighboring pulses thus cannot be distinguished clearly at the receiver. Intersymbol interference (from now on abbreviated as ISI) prevents successful transmission of information.

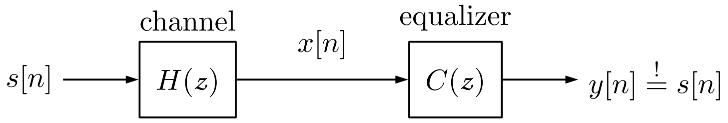


Figure 3.1: Signal flow chain of channel and equalizer (zero-forcing idea)

The intention of an equalizer is to reverse the time-smearing so that in the end the single pulse is confined to its time slot again, as are all subsequent pulses. Figure 3.1 shows the signal flow of such a cascaded system structure of channel and equalizer. The condition to remove ISI given at the output $y[n] = s[n]$ is rather strict. It is intended to convey the general idea of the concept – which ultimately leads to the approach of *zero-forcing* in Section 3.1.

Another obstacle encountered in practice is noise in the transmission. It is usually considered an independent factor of disturbance, meaning the overall disturbance can be split into an ISI component and a noise component. Zero-forcing only addresses the ISI component – and yields suboptimal results in the presence of noise. The advanced idea of simultaneously equalizing both ISI *and* noise leads to the *minimum mean-square error* (MMSE) approach discussed in Section 3.2. This concept is the basis for comprehending – and proceeding towards – *blind* equalization, which is a key topic in this thesis. Several famous approaches will be cited and compared in Section 3.3.1. General remarks on common properties and algorithm convergence are given in Sections 3.3.2 and 3.3.3. Section 3.4 introduces another perspective on blind equalization based on optimal nonlinear filter theory (Bayesian theory) which leads to the "W_{new}" algorithm [37]. Finally in Section 3.5, a new algorithm is designed which matches the requirements posed in this thesis by combining and modifying several of the mentioned equalization concepts. Its relation to CMA (constant modulus algorithm, see Section 3.3.1) is shown.

3.1 Zero-Forcing

The term *zero-forcing* expresses the intention of forcing the transmission error due to ISI to zero at the output. To this end, the mathematical condition can be formulated in multiple equivalent ways. The frequency domain (z-Domain) formulation is the most compact one: Given a channel $H(z)$ with an input signal $S(z)$ (see Figure 3.1), the channel output $X(z)$ is computed by multiplication

$$X(z) = S(z) \cdot H(z). \tag{3.1}$$

In order to single out the original signal $S(z)$ from this equation again, the factor $H(z)$ needs to be eliminated. Evidently, division achieves the goal

$$Y(z) = X(z) \cdot \frac{1}{H(z)} = S(z) \cdot H(z) \cdot \frac{1}{H(z)} = S(z) \tag{3.2}$$

where $Y(z)$ is the output and $\frac{1}{H(z)} = H^{-1}(z)$ is the *inverse* of the given channel. This operation of multiplying the inverse can be seen as appending a second “channel-like” system block *after* the channel (analogous to $C(z)$ in Figure 3.1). The zero-forcing condition can thus be stated as a requirement on the channel $H(z)$

$$H(z) \cdot H^{-1}(z) \overset{!}{=} 1 \tag{3.3}$$

This condition may look simple, but there is more to it than meets the eye. First and foremost, the inverse $H^{-1}(z)$ needs to exist. Oppenheim [38] sets the requirements for the existence of such an inverse (in the context of LTI systems): Assume the system $H(z)$ has a ZPK form as stated in (2.19) in Chapter 2. The zeros are defined by the numerator terms, the poles by the denominator. When the system is inverted, numerator and denominator are swapped. The former zeros thus become poles, the former poles become zeros respectively.

- For the inverse to exist the regions of convergence in the z-plane of $H(z)$ and $H^{-1}(z)$ must overlap. This is the general requirement, without putting further constraints on the inverse.
- For the inverse to additionally be causal (assuming $H(z)$ itself is), the region of convergence must be chosen to be *outside* of the circle defined by the zero of $H(z)$ with the greatest radius.
- For the inverse to additionally be stable, the region of convergence must include the unit circle.

To summarize the core statement, a linear, time-invariant, stable and causal system $H(z)$ has a stable and causal inverse only if all zeros and all poles of $H(z)$ lie inside the unit circle. Such a system is called *minimum-phase system.* As mentioned at the end of Chapter 2, the modeled RFID system does not possess the minimum-phase property. As can be seen in Figure 2.20, one zero lies outside the unit circle. Oppenheim [38] introduces a remedying system decomposition

$$H(z) = H_{min}(z) \cdot H_{ap}(z) \tag{3.4}$$

splitting the system into a minimum-phase part $H_{min}(z)$ containing those zeros and poles inside the unit circle, and an allpass part $H_{ap}(z)$ which – in case of the RFID model – takes care of the zero outside the unit circle. It is important to note that the magnitude frequency spectrum of

$H(z)$ is identical to that of $H_{min}(z)$. The difference lies in the phase of the spectrum, and this difference is expressed by the *allpass* part. The inverse of the system can thus be approximated by inverting only the minimum-phase part (which is invertable by definition) and leaving the allpass part as it is. Since (in this thesis) this phase difference is due to a single zero, it may be compensated for by a phase rotation (see Section 4.2.4).

The zero-forcing condition of (3.3) is relaxed usually. The system cascade of $H(z)$ and $H^{-1}(z)$ does not have to be unity exactly. Instead, it may comprise of a gain constant a_{ZF} and may also be delayed in time by an integer delay of Δ

$$H(z) \cdot H^{-1}(z) \overset{!}{=} a_{ZF} \cdot z^{-\Delta} \tag{3.5}$$

Translated to time-domain, the ZF criterion is defined as

$$h[n] * h_{inv}[n] \overset{!}{=} a_{ZF} \cdot \delta[n-\Delta] \tag{3.6}$$

where $h[n]$ and $h_{inv}[n]$ are the impulse responses of $H(z)$ and $H^{-1}(z)$ respectively, $*$ denotes convolution and $\delta[n]$ is the Kronecker delta. Set in relation to (3.2), the zero-forcing condition on the output $y[n]$ is

$$y[n] = s[n] * (a_{ZF} \cdot \delta[n-\Delta]) = a_{ZF} \cdot s[n-\Delta] \tag{3.7}$$

meaning the input signal $s[n]$ is reconstructed at the output up to a scaling gain factor a_{ZF} and a sample delay Δ.

3.2 Minimum Mean-Square Error

With additive white Gaussian noise $w[n]$ (AWGN) introduced in Figure 3.2, a certain degree of randomness propagates through the flow graph. Since convolution ($*$) is a linear operation, the influence of $w[n]$ can be singled out as a separate entity $\tilde{w}[n]$ at the output:

$$\begin{aligned} y[n] &= r[n] * c[n] \\ &= (x[n] + w[n]) * c[n] \\ &= x[n] * c[n] && + \quad w[n] * c[n] \\ &= (s[n] * h[n]) * c[n] && + \quad \tilde{w}[n] \\ &= s[n] * (h[n] * c[n]) && + \quad \tilde{w}[n] \\ &\overset{(ZF)}{=} a_{ZF} \cdot s[n-\Delta] && + \quad \tilde{w}[n] \end{aligned} \tag{3.8}$$

where $c[n]$ is required to be equal to the inverse system $h_{inv}[n]$ in the last equation line. As can be observed, the additive noise is convolved with the equaizer $c[n]$ (aligned to the right). If the equalizer is adjusted according to the zero-forcing criterion, the term $\tilde{w}[n]$ is very likely to impair the reconstruction of $s[n-\Delta]$, because the applied equalizer $c[n] = h_{inv}[n]$ may cancel the channel ISI perfectly, but the effect on the noise can be arbitrary. In the best case, the equalizer attenuates the noise while $s[n-\Delta]$ can be reconstructed. In the worst case, however,

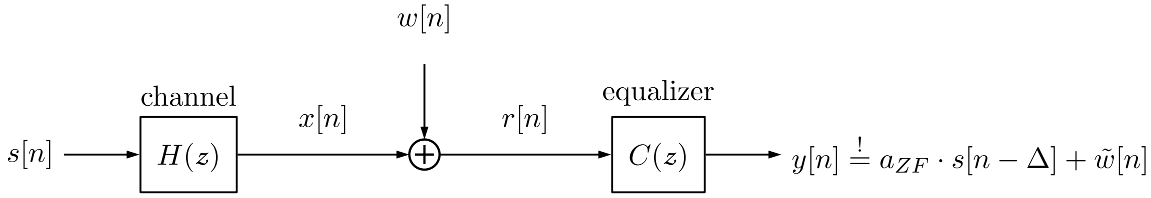


Figure 3.2: Signal flow chain of channel and equalizer under AWGN

the noise may be unintentionally boosted by the response of $c[n]$ (at certain frequencies), so that – even if the $s[n-\Delta]$ part is successfully resolved – the noise $\tilde{w}[n]$ completely overpowers it, rendering it unrecognizable.

3.2.1 MSE criterion

The MSE criterion attempts to achieve the just mentioned "best case". It lowers the requirements on $c[n]$ such that the inverse channel needs to be merely approximated (instead of being exact). This way, the ISI is not forced to *zero*, but to *almost zero*, offering an opportunity to find a similar equalizer $c_{new}[n]$ which possibly will not enhance the noise, but reduce it to a minimum instead. The MSE approach thus looks for a joint optimal solution regarding ISI and noise.

MSE stands for *mean squared error*. This error is defined to be the difference between the equalizer output $y[n]$ and a reference signal $d[n]$ which represents the *desired* output – i.e., what $y[n]$ should look like. This error – on its own – can be expected to fluctuate, especially in the beginning when $y[n]$ and $d[n]$ are still very different. The smart choice is to minimize not the error directly, but to minimize this fluctuation, which is called *variance* in probability theory. It is defined as the expected value (mean value) of the squared error:

$$\Psi = \mathrm{E}\big\{e^2[n]\big\} = \mathrm{E}\big\{(d[n]-y[n])^2\big\} \tag{3.9}$$

In order to find the minimum-MSE solution, the derivative (or gradient) of the MSE *cost function* Ψ has to be computed and then set to zero. In the mathematical field of analysis, this is the standard procedure for finding the minima (and maxima) of the cost function.

$$\boldsymbol{c}_{opt} = \arg\min_{\boldsymbol{c}} \; \mathrm{E}\big\{e^2[n]\big\} \quad \Rightarrow \quad \frac{\partial}{\partial \boldsymbol{c}}\Psi(\boldsymbol{c}) \overset{!}{=} 0 \tag{3.10}$$

The cost is minimized with respect to the equalizer response $\boldsymbol{c}$ as indicated in (3.10), and is further explained in the next section (vector-matrix notation). The foundation for the solution of this problem was established in 1931 by Wiener and Hopf:[20]

$$\boldsymbol{c}_{opt} = \boldsymbol{R}^{-1}\boldsymbol{p} \tag{3.11}$$

Their methodology is clarified in the following section, along with an iterative technique towards the solution.

---

[20] The optimal equalizer coefficients $\boldsymbol{c}$ are (non-iteratively) computed by direct inversion of the autocorrelation matrix $\boldsymbol{R}$. The result is called a "Wiener filter". If $\boldsymbol{R}$ is not invertible, the Moore-Penrose pseudoinverse is a popular choice of approximation. It finds the unique solution with the smallest Euclidean norm among all possible solutions.

3.2.2 Steepest descent

A neat way to solve the minimization problem numerically is the method of steepest descent. To this end, the cost function gradient is determined and then computed iteratively until it reaches the desired minimum. This method works perfectly fine, because the cost function is quadratic – it always has a single unique extremum; the derivative (or gradient) becomes zero in one point only.

To demonstrate the steepest descent, it is helpful to introduce vector-matrix notation. This notation will remain relevant for the stochastic gradient descent (SGD) as well as for the blind equalizers appearing in Section 3.3. The equalizer is chosen to have a certain number L of coefficients, which constitute a vector $\boldsymbol{c}[n] = [c_1 \quad c_2 \quad c_3 \quad ... \quad c_L]^T$. The $[n]$ indicates that the coefficients change over time, the ^{T} indicates vector transpose if $\boldsymbol{c} \in \mathbb{R}$, otherwise ($\in \mathbb{C}$) it must be replaced by Hermitian transpose (^{H}). The received signal is also written as a vector, which contains L "past" samples $\boldsymbol{r}[n] = [\, r[n] \quad r[n-1] \quad r[n-2] \quad ... \quad r[n-(L-1)] \,]^T$ relative to the current n, so that both vectors have the same length. The convolution sum yielding the output $y[n]$ at each discrete time instant n can now be expressed as a scalar product of $\boldsymbol{c}^T[n]\boldsymbol{r}[n]$.[21] Substituting the newly established notation into (3.9) gives

$$\begin{aligned}\Psi(\boldsymbol{c}) &= \mathrm{E}\big\{(d[n] - \boldsymbol{c}^T\boldsymbol{r})^2\big\} \\ &= \mathrm{E}\big\{d^2[n]\big\} - 2\mathrm{E}\big\{d[n]\cdot\boldsymbol{c}^T\boldsymbol{r}\big\} \quad + \mathrm{E}\big\{(\boldsymbol{c}^T\boldsymbol{r})^2\big\} \\ &= \quad \sigma_d^2 \qquad - 2\boldsymbol{c}^T\cdot\mathrm{E}\{d[n]\cdot\boldsymbol{r}\} + \boldsymbol{c}^T\mathrm{E}\big\{\boldsymbol{r}\boldsymbol{r}^T\big\}\,\boldsymbol{c} \\ &= \quad \sigma_d^2 \qquad - 2\boldsymbol{c}^T\cdot \qquad \boldsymbol{p} \qquad + \quad \boldsymbol{c}^T\,\boldsymbol{R}\,\boldsymbol{c}\end{aligned} \tag{3.12}$$

where $\boldsymbol{R}$ is the autocorrelation matrix of $\boldsymbol{r}[n]$. The vector $\boldsymbol{p}$ can be regarded as the crosscorrelation between $d[n]$ and all past samples stored in $\boldsymbol{r}[n]$. The variables $\boldsymbol{p}$ and $\boldsymbol{R}$ are assumed to be known, exact, and therefore time-invariant (stationary process assumption).[22] The gradient with regard to $\boldsymbol{c}$ is given by

$$\begin{aligned}\nabla_{\boldsymbol{c}}\Psi(\boldsymbol{c}) &= 0 - 2\boldsymbol{p} + 2\boldsymbol{R}\boldsymbol{c} \\ &= 2(\boldsymbol{R}\boldsymbol{c} - \boldsymbol{p})\end{aligned} \tag{3.13}$$

To find the minimum, the gradient is set to zero. The factor 2 can be omitted. The direct solution is the Wiener-Hopf equation already stated in (3.11). It requires inversion of the autocorrelation matrix $\boldsymbol{R}$. To circumvent this, the steepest descent method can be used. An iterative update equation is introduced, which adjusts the coefficients in $\boldsymbol{c}$ by taking a "step" in direction of the negative gradient, yielding the new coefficients $\boldsymbol{c}[n+1]$:

$$\boldsymbol{c}[n+1] = \boldsymbol{c}[n] + \mu\cdot\Big(-\nabla_{\boldsymbol{c}}\Psi(\boldsymbol{c}[n])\Big) = \boldsymbol{c}[n] + \mu\cdot\Big(\boldsymbol{p} - \boldsymbol{R}\,\boldsymbol{c}[n]\Big) \tag{3.14}$$

where μ is the stepsize parameter. Figure 3.3 visualizes the process of finding the solution iteratively, for the case of an equalizer with two coefficients. Starting at some point at the top of the basin, the coefficients are updated following the black arrows marking the path of steepest descent. The solution is reached when the bottom of the basin is reached. The axis ticks are intentionally left blank, because the solution values in each case depend on various parameters. Detailed information on this topic is found in [26].

---

[21] For the sake of clarity, the time index $[n]$ is omitted after the vector variables $\boldsymbol{c}$ and $\boldsymbol{r}$ in (3.12),(3.13) and (3.15); it is explicitly stated in (3.14) to emphasize the iterative nature of the steepest descent method.

[22] as opposed to (3.15a) and (3.15b), where the instantaneous estimates $\hat{\boldsymbol{p}}$ and $\hat{\boldsymbol{R}}$ are in fact time-*variant*

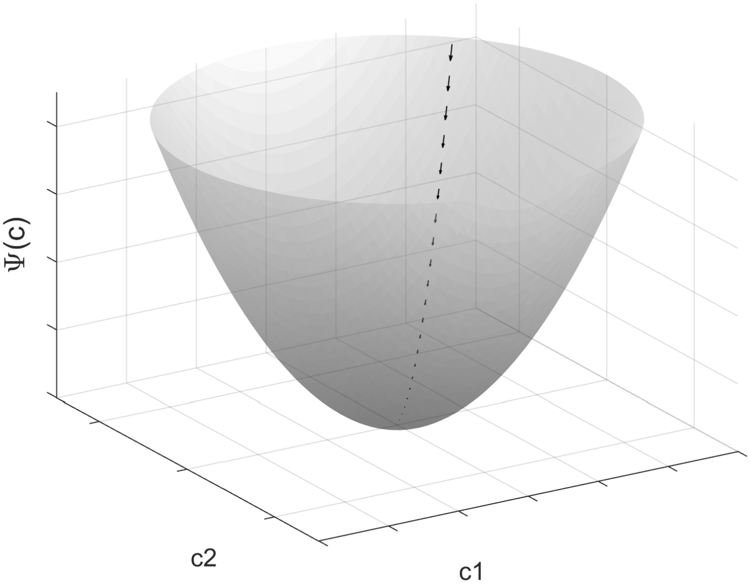


Figure 3.3: MSE cost function surface

3.2.3 LMS Algorithm

The LMS algorithm was first used by Widrow and Hoff (1960) in the context of adaptive neural networks. Since then it has become one of the most widely used adaptive algorithms due to its simplicity and effectiveness. It is frequently used to compare the performance of rivalling algorithms – likewise in this thesis with the blind algorithms (see Chapter 4).

The LMS algorithm belongs to the family of stochastic gradient algorithms. It differs from the steepest descent method stated in the previous section such that the gradient is not computed deterministically ($\boldsymbol{Rc} - \boldsymbol{p}$), but is instead replaced by a gradient *estimator*. Due to the variance of this estimator, the exact solution is only approximated (excess mean-square error, comp. [26]). The exact gradient measures $\boldsymbol{R}$ and $\boldsymbol{p}$ are replaced by *instantaneous estimates* based on the sample values that are available at each time instant ($\in \mathbb{C}$, * denotes complex conjugation, ^{H} Hermitian transpose):

$$\hat{\boldsymbol{R}} = \boldsymbol{r}\boldsymbol{r}^H \quad \overset{?}{\approx} \quad \mathrm{E}\{\boldsymbol{r}\boldsymbol{r}^H\} \tag{3.15a}$$

$$\hat{\boldsymbol{p}} = d^*[n] \cdot \boldsymbol{r} \quad \overset{?}{\approx} \quad \mathrm{E}\{d^*[n] \cdot \boldsymbol{r}\} \tag{3.15b}$$

Computational requirements drop significantly due to this simplification (estimating the true autocorrelation matrix or crosscorrelation vector would be computationally expensive). In fact, the computation is feasible in real-time, because few additional operations are needed compared to a static FIR filter[23]. The performance of LMS is quite satisfactory, because it is still able to closely reach the optimum solution – at the cost of a possible “zig-zag” path towards the solution (comp. Figure 3.3, where the path is straight), and at the cost of hovering around the minimum forever because of non-vanishing coefficient fluctuation. Still, the LMS solution can be regarded as very successful. It can achieve the desired joint optimization of ISI and noise.

---

[23] transversal filter structure; modifications like lattices also possible; more information in [26, p.5, 6, 218]

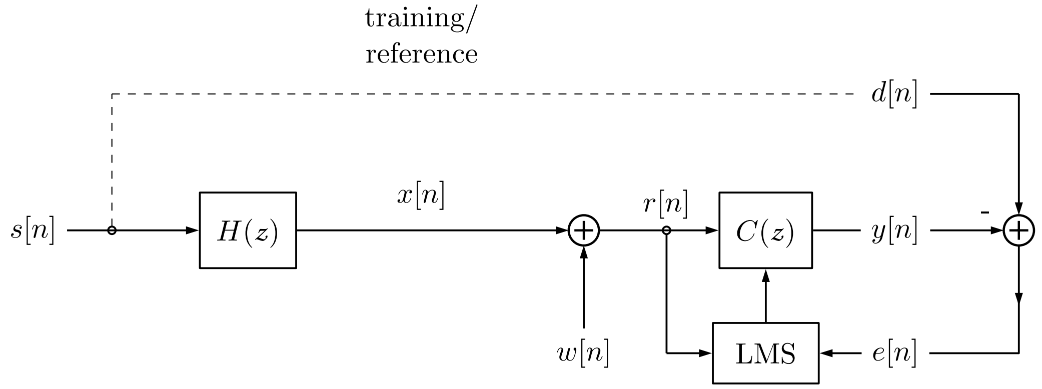


Figure 3.4: Signal flow graph of adaptive LMS channel equalization

LMS algorithm	
0. Initialize $\boldsymbol{c}[n]$	
1. Filter output:	$y[n] = \boldsymbol{c}^H[n]\boldsymbol{r}[n]$
2. Current error:	$e[n] = d[n] - y[n]$
3. Coefficient update:	$\boldsymbol{c}[n+1] = \boldsymbol{c}[n] + \mu \cdot \boldsymbol{r}[n] \cdot e^*[n]$

Table 3.1: LMS algorithm procedure and equations

Back in the RFID context again, the LMS has a major drawback making it inapplicable as an equalizer under the ISO-14443 standard: The reference signal $d[n]$ – which is often called **training sequence** – needs to be transmitted repeatedly and to be known at the receiver. The information rate drops to undesirable levels. The short time frame, in which the card can send its message before energy runs out, turns to the disfavor of LMS. Additionally, every card already out there would need to be updated to know the training sequence in order to use the advantages of the LMS equalizer.

The LMS algorithm is put to good use in this thesis after all: First and foremost as a benchmark for the blind algorithms (see Section 4.3), secondly in an offline manner to pre-compute inverses of the RFID channel model that has been developed in Chapter 2. The graph shown in Figure 3.4 is therefore changed slightly to facilitate inverse system identification. In the offline case, the reference sequence $d[n]$ can be chosen optimally to compute quite accurate inverses for all k. Figure 3.5 shows the zeros of these identified inverses. Comparing it to Figure 2.20, clearly the zeros cancelling the corresponding channel poles have been found. The remaining two zeros attempt to make up for the non-minimumphase behaviour of the RFID channel. This is an important intermediate step: The inverses shown here will be relevant for the initialization of the devised blind equalizer (see Section 4.2.4).

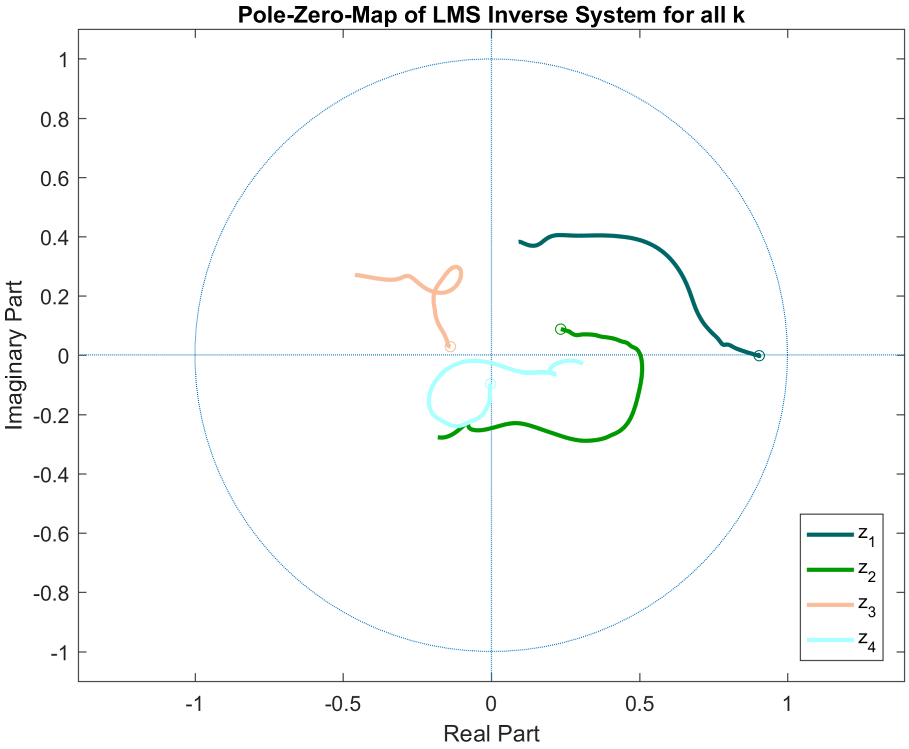


Figure 3.5: Zeros of the LMS inverse system approximation

3.3 Blind equalization

Blind equalization is closely related to the classical adaptive methods described previously. The general intention of equalizing the communications channel (ISI and noise) stays exactly the same. Actually, there is just this tiny step that makes an equalizer *blind*: the restriction that the **training sequence** $d[n]$ (see Figure 3.4) is not to be used. This means that the desired output has to be computed *without any direct reference* to what the perfect output should look like. Merely the signals measurable at the receiver are available to solve this task.

This simple restriction on $d[n]$ opens an entire universe of different approaches. Blind methods of arbitrary complexity exist in the literature. Blind techniques in neural networks and machine learning are examples of using massive amounts of data and computationally expensive algorithms to arrive at a solution. Other approaches make use of explicit higher-order statistics, which also rely on great sample amounts to be successful. The blind algorithms considered in this thesis are – in contrast – expected to perform well given a bare minimum of samples and processing operations. Efficiency, low complexity, and real-time compatibility are the features of interest here.

Algorithms closely related to LMS are an attractive option. Such algorithms also belong to the family of *stochastic gradient descent* (SGD) methods. The steps 1. and 3. in the LMS algorihm Table 3.1 will thus stay exactly the same for the examined blind algorithms. Merely step 2. needs to be redefined because of the forbidden $d[n]$. A new error signal is needed: a signal that can compensate for the missing $d[n]$ by replacing it with a signal which behaves similar to $d[n]$, but is composed of the available signals ($\boldsymbol{r}[n]$, $y[n]$). Incorporating some useful assumptions on the input signal $s[n]$ (e.g., statistical properties) can also help in this matter.

The basic idea behind all algorithms listed hereafter is to devise a new cost function $\Psi(y[n])$, or equivalently a new *error function* $\Psi'(y[n]) = \nabla_{\boldsymbol{c}}\Psi(y[n])$, which directly defines the gradient (without revisiting a cost function). By a good choice of such a function, the left part of (3.14) applies, so that the blind algorithm can take steps in the direction of the negative gradient – to converge to a solution of successful equalization. Several popular examples of such cost or error functions will be given now (Section 3.3.1), followed by remarks on why these functions are able to achieve equalization (Section 3.3.2) and in which situations they do not work well or not at all (Section 3.3.3). A recommendable overview on blind algorithms and their properties and behaviour is given by Johnson's research group in [13].

3.3.1 Famous blind algorithms

Sato algorithm
This algorithm is the first blind equalisation algorithm to be proposed in 1975 by Yoichi Sato [6]. M-level PAM is assumed as input signal (allowing assumption on the input statistics). It is defined via error function

$$\Psi'_{Sato}(y[n]) = y[n] - \frac{\mathbb{E}\{\ |s[n]|^2\ \}}{\mathbb{E}\{\ |s[n]|\ \}} \cdot sgn(y[n]) \tag{3.16}$$

For binary input (2-level PAM, ±1), the expectation factor is 1, reducing the error to $e[n] = y[n] - sgn(y[n])$, which is the same as comparing the output $y[n]$ to the output of a decision device (slicer) $Q(y[n]) = sgn(y[n])$. For M-level PAM generally, the algorithm can be seen as a sort of more sophisticated slicer. The success of this algorithm depends on the slicer decisions being correct, that is, $sgn(y[n]) = s[n]$ or $s[n-\Delta]$ for all n – or at least for most n. In case of severe ISI, this does not necessarily hold, in which case this algorithm runs into trouble.

Stop-and-Go algorithm
This algorithm proposed in [10] attempts to improve the convergence time of the previously presented approach by imposing a rule about when it is promising to make a step towards convergence and when it is better to keep the current coefficient vector as it is. This rule takes advantage of the fact that the direction towards convergence is more likely to be correct if the signs of two separate error functions are equal at a particular time step n. It thus includes results of more than one blind method in its convergence process. In the paper, the decision-directed (DD) approach and the Sato algorithm are combined:[24]

$$\Psi'_{StopGo}(y[n]) = \frac{1}{2}\left(y[n] - Q(y[n])\ + |y[n] - Q(y[n])| \cdot sgn(y[n] - \frac{\mathbb{E}\{\ |s[n]|^2\ \}}{\mathbb{E}\{\ |s[n]|\ \}} \cdot sgn(y[n])\right) \tag{3.17}$$

This looks a bit confusing, because the two algorithms are consolidated into one function requiring only one update equation without further condition checking. The rewritten (general) form looks like this simple coefficient update rule (involving any two error functions $\Psi'_1,\ \Psi'_2$)

$$\mathbf{c}[n+1] = \begin{cases} \mathbf{c}[n] - \mu\Psi'_1(y[n])\mathbf{x}[n] & if\ \ sgn(\Psi'_1) = sgn(\Psi'_2) \\ \mathbf{c}[n] & if\ \ sgn(\Psi'_1) \neq sgn(\Psi'_2) \end{cases} \tag{3.18}$$

---

[24] Note: Sato and DD are identical in case of 2-PAM input; this algorithm suggests PAM with $X > 2$ levels

BGR algorithms

This family of algorithms is considered to be an extension of Sato's idea. It has been introduced by Benveniste, Goursat and Ruget [8], hence the name (BGR). Their error function (type) is given as

$$\Psi'_{BGR}(y[n]) = f(y[n]) - \frac{\mathbb{E}\{\ f(s[n]) \cdot s[n]\ \}}{\mathbb{E}\{\ |s[n]|\ \}} \cdot sgn(y[n]) \tag{3.19}$$

where $f(x)$ is the novelty. It is defined to be *odd* and *twice differentiable* (except at the origin) with $f''(x) \geq 0$ for all $x \geq 0$. The Sato algorithm can be deduced by setting $f(x) = x$, which fulfills $f''(x) = 0$. Similarly, a certain subset of odd-powered polynomials will fulfill these conditions on the error function. This train of thought links the BGR family to the later presented Godard family algorithms. Paper [8] also states the simple extension of BGR algorithms to the complex signal case (QAM):

$$y[n] \ \rightarrow \ \mathrm{Re}\{y[n]\} + j\ \mathrm{Im}\{y[n]\} \tag{3.20}$$

$$\Psi'_{BGR}(y[n]) = \Psi'_{BGR}(\mathrm{Re}\{y[n]\}) + j\ \Psi'_{BGR}(\mathrm{Im}\{y[n]\}) \tag{3.21}$$

The coefficient update equation is then modified by a complex conjugation

$$\mathbf{c}[n+1] = \mathbf{c}[n] - \Psi'_{BGR}(y[n]) \cdot \mathbf{r}^*[n] \tag{3.22}$$

Constant Modulus (Godard) algorithms

Godard starts off with cost functions in his paper [7], not their derivatives as seen in the previous methods. He denotes his cost functions with a $\mathcal{D}^p$ indicating the underlying idea of dispersion of order p. He points out that for pure phase modulation (M-PSK), it is sufficient to constrain the equaliser output to possess constant amplitude (constant modulus) for successful equalisation. The Godard cost function is of the form

$$\Psi_{Godard}(y[n]) = \frac{1}{2p}\Big(|y[n]|^p - \frac{\mathbb{E}\{\ |s[n]|^{2p}\ \}}{\mathbb{E}\{\ |s[n]|^p\ \}}\Big) \qquad p = 1, 2, \ldots \tag{3.23}$$

For $p = 1$, this cost function is equivalent to the the Sato cost function (the integral of the error function Ψ'_{Sato}):

$$\Psi_{Godard}(y[n])\Big|_{p=1} \ = \ \Psi_{Sato}(y[n]) = \frac{1}{2}\ \Big(\ |y[n]| - \frac{\mathbb{E}\{\ |s[n]|^2\ \}}{\mathbb{E}\{\ |s[n]|\ \}}\ \Big)^2 \tag{3.24}$$

To be consistent with the presentation of previous algorithms, the error function of the Godard algorithms shall be stated here as well:

$$\Psi'_{Godard}(y[n]) = \Big(|y[n]|^p - \frac{\mathbb{E}\{\ |s[n]|^{2p}\ \}}{\mathbb{E}\{\ |s[n]|^p\ \}}\Big) \cdot |y[n]|^{p-2} \cdot y[n] \tag{3.25}$$

The constant modulus condition has the effect that analysed signal portions *not* having a constant modulus (channel influences and noise) are inherently suppressed, while the portion *having* a constant modulus is favoured. Another feature of the Godard algorithms is their invariance towards the phase of the output $y[n]$, as only its absolute value is considered in the cost function. Potential phase rotations or synchronizations need to be taken care of independently. The Godard algorithm for $p = 2$ is the most popular blind SGD algorithm. It is often denoted as CMA – "the" constant modulus algorithm – although the algorithms of other p also exploit the constant modulus property.

Shalvi-Weinstein algorithms

This method [11] is distinguished from the previous algorithms in its explicit use of higher-order statistics (HOS), specifically the *kurtosis*. The overall system impulse response $o[n]$ – which should simply be a single Kronecker delta (fulfilling the zero-forcing condition, see equation 3.6) – is constrained for constant unity power

$$o[n] = h[n] * c[n] \quad with \sum_{n=-\infty}^{\infty} |o[n]|^2 \overset{!}{=} 1 \tag{3.26}$$

If this holds (a normalization step or automatic gain control is required), and if the input signal s[n] is i.i.d., the input and output signal power (variance) are related by

$$\mathbb{E}\{|y[n]|^2\} = \mathbb{E}\{|s[n]|^2\} \quad \cdot \sum_{n=-\infty}^{\infty} |o[n]|^2 \tag{3.27}$$

as are their 4^{th}-order moments (kurtosis)

$$K_y = K_s \cdot \sum_{n=-\infty}^{\infty} |o[n]|^4 \tag{3.28}$$

with the kurtosis defined as (K_s similarly)

$$K_y = \mathbb{E}\{\ |y[n]|^4\ \} - 2\mathbb{E}^2\{\ |y[n]|^2\ \} - |\ \mathbb{E}\{\ y^2[n]\ \}\ |^2 \tag{3.29}$$

The SGD is used in this algorithm as well, but with the intention of *maximising* the kurtosis at the output. Most QAM and PAM signals $s[n]$ exhibit nonzero kurtosis ($K_s \neq 0$), making this property a sufficient criterion for input reconstruction at the output. The output kurtosis can be maximised by maximising the overall system kurtosis $K_o = \sum |o[n]|^4$. For details, refer to [11].

3.3.2 Bussgang property and higher-order statistics

The last listed blind method uses higher-order statistics (the kurtosis) in a direct way. The other methods use HOS implicitly – their error functions are inherently *nonlinear* (absolute value, signum function, polynomials). They count as *static* (memoryless) nonlinearities. In contrast to linear functions, nonlinear functions have the ability to change the probability distribution of its input (which is considered to be a stochastic process in this case).[25] This implies that by choosing an appropriate nonlinear function, the distribution of a message signal $s[n]$ can be enhanced at the output to assist reconstruction.

---

[25] see *Nonlinear Signal Processing VO* scriptum

This leads to the Bussgang property. It is a kind of guideline for ensuring that $s[n]$ can indeed be reconstructed for a specific nonlinear function. The original theorem by J. Bussgang (1952) says the following:

Definition Let $X(t)$ and $Y(t)$ be the input and output random variables (processes) respectively, of a nonlinear operation (function) $g()$, such that X is a zero-mean Gaussian process $X(t) \sim \mathcal{N}_t(0, \sigma_x^2)$ and $Y(t) = g(X(t))$

Bussgang theorem *The cross-correlations of a Gaussian signal – before and after passing through a nonlinear operation – are equal, up to a constant* $\mathbf{C}$ *which solely depends on* $g()$.

$$r_{xy}(\tau) = \mathbf{C} \cdot r_{xx}(\tau) \qquad \text{with} \quad \mathbf{C} = \frac{1}{\sigma_x^3 \sqrt{2\pi}} \int_{-\infty}^{\infty} u \cdot g(u) \cdot e^{-\frac{u^2}{2\sigma^2}} du \tag{3.30}$$

Barrett and Lampard (1955) found that a great number of stochastic processes (beside the Gaussian ones described here) also possess the Bussgang property and extended the Bussgang theorem to all stochastic processes with exponentially decaying auto-correlation functions r_{xx} (as the significant peak is still at $r_{xx}(0)$). This includes Uniform and Laplace distributions, or more generally everything in between labeled “sub- and supergaussian distributions”. They are classified by their kurtosis: The Gaussian distribution has kurtosis 0, the associated distributions have either smaller or greater kurtosis, respectively.[26] The important thing to note here is that distributions of common signal constellations (like M-PAM, M-PSK, M-QAM) can be ranked among these extended distributions.

Blind algorithms designed for these constellations are then called “Bussgang algorithms” because they fulfil (or at least almost fulfil) the Bussgang theorem when convergence is completed (here in discrete time):

$$\mathrm{E}\{y[n]y[n-\tau]\} \simeq \mathrm{E}\{y[n]g(y[n-\tau])\} \tag{3.31}$$

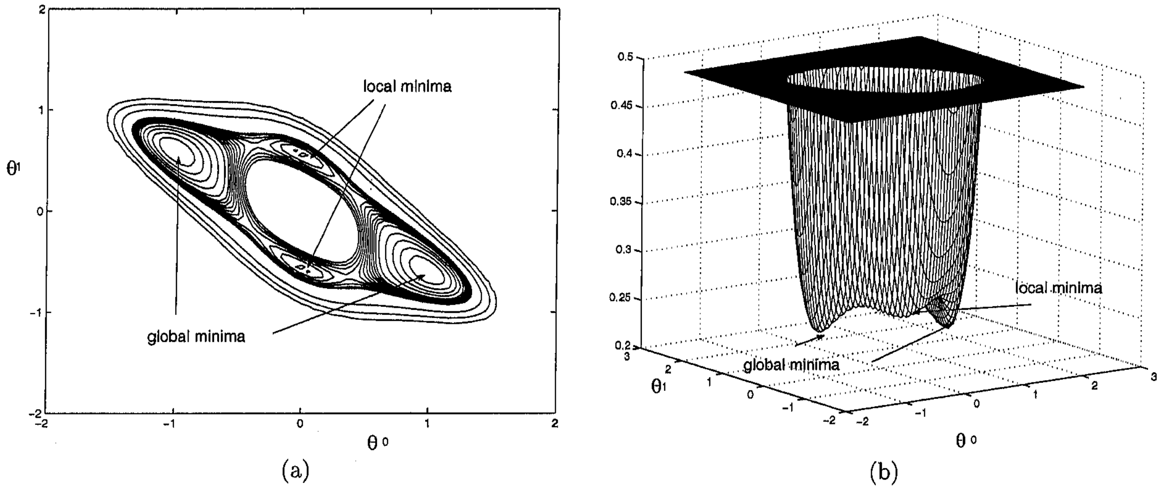
This implies that, assuming $s[n]$ has been reconstructed successfully ($g(y[n]) \approx \hat{s}[n-\Delta]$), the output signal (process) attains the statistical properties inherent to the particular chosen constellation producing the (unknown) input process. The convergence process can thus be interpreted as *encouraging* the distorted signal to transform (via equalizer) into a signal that likely belongs to such an underlying constellation. Ding [12] gives details by means of a MAP estimator approach. He also points out that the blind algorithms (from section 3.3.1) are sub-optimal versions related to this MAP estimator. This remark is also relevant for the Bayesian approach in Section 3.4.

3.3.3 Initialization and convergence

The term “sub-optimal” gives the cue for this section, which addresses the challenges inherent to blind equalizers. The main concern is whether convergence to a valid solution can be guaranteed. This question arises, because the cost functions of the previously listed algorithms are non-convex. On the opposite, the LMS cost function *is* convex, which means its cost surface (see Figure 3.3) has a single “basin” with a clearly defined bottom (which portraits the minimum, the point where the solution is achieved). Non-convex cost surfaces show multiple “basins” (see Figure 3.6) – not all of which correspond to a valid solution (valid = *global* minimum). If, during the iterations of the algorithm, a step is accidentally made in the wrong direction (due

---

[26] Table 2 in [13] shows the kurtosis values for all customary constellations.



(a)

(b)

Figure 3.6: CMA cost function surface for noiseless 2-PAM over AR(1) channel and a two-tap equalizer (L = 2 overparametrized) showing length-dependent local minima; the two global minima are due to the CMA's phase ambiguity; Direct copy of [12]

to the additive noise, or because the current frame of received samples is particularly distorted by ISI at this instant), the algorithm may end up in the wrong "basin" (*local* minimum). In consequence, the algorithm stops (because it thinks it has converged) and the equalizer keeps applying coefficients that do not help reconstructing the input message $s[n]$.
Based on findings in [12], [13] and [39], a short summary is given here about how such non-convex cost surfaces behave and which parameters influence their behavior.

Local minima may appear due to

- the cost function and/or error function directly. They appear even if all other parameters fit perfectly – i.e., even if a fixed (static) equalizer could do the job. Example: The Sato cost will show local minima in the situation of 2-PAM (± 1) i.i.d. zero mean input to an AR(n) channel, with an FIR equalizer perfectly achieving the MA(n) process that completely cancels the AR(n) channel effect, and no noise (AWGN) present. Stop-and-Go and BGR algorithms also show this behaviour. These minima are labeled **cost-dependent local minima**. CMA and the Shalvi-Weinstein algorithm do not have minima of this type. They will converge *globally* in the situation given above.
- the FIR filter *not* having adequate length L. This issue includes consideration about the invertibility of the channel (given in Section 3.1). In a practical scenario the channel may be varying over time, the length of the inverse may thus also vary accordingly. The problem is most prominent for channels predominantly characterized by zeros (MA-processes) – which is luckily not the case for the modeled RFID channel (the poles = AR-process are dominant). All of the listed algorithms show these **length-dependent local minima**. They vanish only in the theoretical case of the infinite-order Wiener filter.[a]
- the use of *real-valued* constellations on *complex-valued* channels and/or equalizers. This is a problem crucial to the investigated RFID channel in this thesis. Literature shows the discovery of this problem but offers no insights in how to combat it. During the choice of algorithms for the RFID channel in Section 3.5 a heuristic proposal is made to reduce the influence of these **number-set-dependent** minima.

---

[a] Paper [40] attempts to eliminate length-dependent minima by making use of nonlinearities *with* memory.

The following items give an idea of how stochastic gradient family (SGD) algorithms behave on non-convex cost surfaces. Mentioned extrema can be either minima or maxima. Saddles are treated separately.

- At any point on the surface far away from an extremum the gradient (first derivative) at that point determines the speed of convergence.
- In vicinity of an extremum rather the second derivative (local curvature) determines the speed of convergence.
- Excess steady-state error occurs if the cost function value at an extremum is nonzero.
- Multimodal surfaces may have extrema with different cost function values; convergence to the correct minimum vitally depends on *where* on the surface the journey for convergence starts.
- Starting at different points on the surface may result in different solutions.
- Saddle points complicate convergence. If such a point lies on the path to convergence, the speed of convergence may become arbitrarily slow, depending on how close the path passes the saddle. *Both* first and second derivative approach zero in vicinity of a saddle (to be more exact, the gradient approaches zero and the Hessian matrix is indefinite).

The choice of a starting point is called *initialization.* In contrast to the LMS – which can converge fine even with all equalizer coefficients set to zero – the performance of blind equalizers depends on correct initialization. Common tactics found in literature are the “center tap” initialization, where the equalizer length L is odd and all coefficients are set to zero except the middle one. Another suggestion is called “tap-anchoring”, where it is attempted to fix the center coefficient at a specifically chosen delay. Gain estimation and control is needed for this method. However, these ideas can only be seen as general suggestions for initialization – they are not universally valid.

Application-specific initialization is likely to perform better. All insights gained in Chapter 2 about the RFID system and especially the potential *inverses* of it presented in Section 3.2.3 will prove useful in this matter.

Some final selected considerations regarding blind equalizers are presented below:

- In the presence of noise (AWGN), the cost surface becomes more shallow. Locations of minima (also the global ones) change according to noise intensity, but their regions of attraction stay similar. Under high noise it is possible to escape from local minima – but also to be diverted from a global minimum.
- Undermodeling of the equalizer (L too small) results in residual ISI (which is usually accepted in practice if sufficiently low). However, the residual also causes deformations of the surface, depending on the magnitude of the residual.
- Channels may vary over time. If this variation is sufficiently slow, the (converged) blind equalizer is able to track channel changes with a speed depending on the local curvature (second derivative).
- The stepsize parameter μ needs to be adjusted according to a trade-off between convergence speed (transient behaviour) and excess error (steady state behaviour). Stepsize $\mu \lll$ means slow convergence but a solution “on-the-spot”, while $\mu \ggg$ means fast convergence but never-ending big steps in the vicinity of the solution (see also Section 4.2.1).

3.4 Bayesian approach

The conclusion on the Bussgang property in Section 3.3.2 implies that equalization is successful when the input signal statistics match those at the output, and that the “driving force” behind the Bussgang algorithms encourages this to happen. This has, however, not been proven universally – it has simply been found that the Bussgang property holds after successful convergence.

The logical step – to explicitly drive the output towards the correct statistics – has been explored in [37]. In this section, this approach is summarized, because it is less restrictive than other blind algorithms that have been short-listed for this thesis, and it has proved to outperform CMA (in the referenced paper, as well as for the RFID channel, see Chapter 4).

The assumptions made in the approach match the requirements of the RFID application:

1. non-minimumphase LTI channel with zeros $|z| \neq 1$ only
2. linear transversal equalizer (adaptive FIR)
3. AWGN noise
4. no assumption on the input probability distribution $\Pi(s)$ (PDF/PMF) except that it has even symmetry
5. the input is i.i.d. and zero-mean.

Furthermore, the MSE approach is maintained (which includes zero-forcing). The idea is restated from a slightly new perspective: the output is split into three additive terms

$$y[n] = a_{ZF} \cdot s[n-\Delta] \quad + \quad \xi[n] * s[n] \quad + \quad \tilde{w}[n] \tag{3.32}$$

where the first term is the ZF solution (see (3.6) and (3.7)), the second term denotes the residual ISI where $\xi[n]$ is the difference (misalignment) of the coefficients $c_{current}[n]$ and $c_{opt}[n]$, and the third term is the shaped noise as in (3.8). Second and third term are intended to vanish with proceeding convergence.

The closed-form approach in the paper [37] explicitly uses the conditional expectation integral (which is the optimal point estimator in a statistical sense) and invokes Bayes’ law to form the true Bayes’ estimator

$$\mathrm{E}\{s|y\} = \frac{\int_{-\infty}^{\infty} s \cdot \Pi(y|s) \cdot \Pi(s) ds}{\int_{-\infty}^{\infty} \Pi(y|s) \cdot \Pi(s) ds} \tag{3.33}$$

where $\mathrm{E}\{s|y\}$ estimates the mean of the true posterior probability distribution $\Pi(s|y)$, $\Pi(y|s)$ is the sampling distribution (also called “measurement distribution” or “likelihood”), which is based on the measured output samples y, given the input assumptions on s (conditional probability). The denominator term is called “evidence” or “marginal likelihood”. In the next step, the prior information ($\Pi(s)$, the distribution of the input) is assumed to be unknown – which means no simplifications towards MAP or ML – it is instead approximated by the maximum entropy method, under the constraint that a sufficient number of moments $m_i = \int s^i \cdot \Pi(s) ds$ of the input are known. Lagrange multipliers are used to solve the approximation of $\Pi(s)$.

With high computational complexity (offline), the full Bayes estimator is applied in the algorithm error as a sophisticated replacement for $d[n]$ (the reference or training signal):

$$\boldsymbol{c}[n+1] = \boldsymbol{c}[n] - \mu \cdot \Big(\mathrm{E}\{s|y\} \, \frac{y[n] \cdot \mathrm{E}\{s|y\}}{\mathrm{E}\{y^2[n]\}} - y[n] \Big) \cdot \boldsymbol{r}^*[n] \tag{3.34}$$

The advantage of this approach is that the residual ISI is estimated along with the input distribution. The high-complexity algorithm is then simplified by assuming a low enough level of ISI and presuming $\mathrm{E}\big\{y^2[n]\big\} \approx \mathrm{E}\big\{s^2[n]\big\}$ so the output variance does not need to be computed for each step. Also, the conditional expectation is replaced by instantaneous estimates (comp. Section 3.2.3). Then the derived algorithm (SGD type) is suited for real-time application – although it is now suboptimal due to the simplifications.

$$e[n] = \Big(\frac{y^3[n]}{\mathrm{E}\{s^2[n]\}} - y[n] \Big) \quad \in \mathbb{R} \tag{3.35}$$

$$\overset{or}{\equiv} \Big(\frac{\mathrm{Re}\{y[n]\}^3}{\mathrm{E}\Big\{\mathrm{Re}\{s[n]\}^2\Big\}} + j \, \frac{\mathrm{Im}\{y[n]\}^3}{\mathrm{E}\Big\{\mathrm{Im}\{s[n]\}^2\Big\}} - y[n] \Big) \quad \in \mathbb{C} \tag{3.36}$$

3.5 Proposed new algorithm

The algorithm in the previous Section 3.4 (it is plainly called “W_{new}” in [37]; they use the letter W for the error $e[n]$) is assessed to be the best choice for the given RFID channel equalization problem (see Chapter 4). This is mainly due to its clear derivation and its flexibility towards any chosen constellation.

These attractive features triggered further investigations concerning the challenge of eliminating *number-set-dependent local minima* (see Section 3.3.3), which are inherent to the given RFID task. This detail seems to have attracted no attention of research as of yet. The reason for this may be explained rather simply: If the application assumed (i.e., implemented) the channel and/or the equalizer to be *complex-valued*, it would seem irrational not to choose a complex alphabet also. Otherwise (if a real-valued alphabet is preferred) the alternative of omitting complex-valued realizations altogether saves computations, thus increases efficiency, which is why a purely *real-valued* solution will then often overrule the more complicated “mixed-number-set” alternative.

In case of the given RFID system, the purely real-valued alternative is not viable. The nature of the complex-valued RFID baseband channel (see Chapter 2) prohibits the mere omission of the imaginary parts. Equalizing the real parts only is bound to fail. Consequently, the equalizer is required to be complex-valued as well. The feature that makes this RFID application special is the intentional choice of BPSK (which is real-valued). The ISO-14443 standard endorses this choice. RFID communication is a much more fragile procedure compared to other communication fields such as digital radio or television. Robustness is of highest priority – the message must be transmittable, regardless of the type of card(-antenna). And out of all known constellations, BPSK is the most proficient one in terms of noise distortion robustness, and power efficiency.

Since the algorithm is *blind* it has no means of knowing whether the input might be real- or complex-valued. Because it is necessarily implemented *complex-valued* in this RFID case, it will keep its options open. So if an adverse combination of input plus channel occurs at any time instant, the blind equalizer tends to interpret the incoming (erratically complex-valued) signal amplitudes as originating from QPSK or QAM (complex-valued) alphabets instead of duly real-valued BPSK.

A modification is proposed in this thesis to overcome this detrimental tendency. The "W_{new}" algorithm in (3.36) inspired this modification, because real and imaginary parts are treated separately.

MODIFICATION 1

Following the argument that the received signal $r[n]$ and the equalizer output $y[n]$ are complex-valued by definition, the error computation should be "made aware" that the *expected* output is in fact real-valued. Without mathematical proof, this is heuristically accomplished by comparing the nonlinear estimator output against the real part of $y[n]$ instead of a general $y[n] \in \mathbb{C}$.

$$e[n] = g(y[n]) - \mathrm{Re}\{y[n]\} \tag{3.37}$$

Section 4.4 will show experimental verification that this modification fulfils the purpose of accomodating both number sets.

The obvious complementary modification has been found to be infeasible: omitting or changing the imaginary part of the nonlinear estimator in (3.36). If omitted, the estimator deteriorates drastically because the imaginary part is not accounted for in the estimation anymore. If changed, the given purpose can at least be approached: It is known for BPSK input that $\mathrm{E}\{\mathrm{Im}\{s[n]\}^2\} = 0$. The denominator of the imaginary term in (3.36) is then 0, which renders the overall term inifinite. The imaginary infinity is then compared against whatever (small) imaginary value the current output signal $y[n]$ has – which means the imaginary part of the whole error signal becomes a constant (close to infinity) rendering those small changes negligible. Mathematically, the outcome of the error modified this way will approach perfect BPSK "comprehension" as $\mathrm{E}\{\mathrm{Im}\{s[n]\}^2\} \longrightarrow 0$. Numerically, this idea is obviously highly problematic.

The values of $\mathrm{E}\{\mathrm{Re}\{s[n]\}^2\}$ and $\mathrm{E}\{\mathrm{Im}\{s[n]\}^2\}$ are actually constellation-dependent; they are fixed constants during algorithm iteration. For BPSK

$$\begin{aligned}&\mathrm{E}\{s^2[n]\} = \mathrm{E}\{\mathrm{Re}\{s[n]\}^2\} = 1\\&\mathrm{E}\{\mathrm{Im}\{s[n]\}^2\} = 0\end{aligned}$$

The proposed MODIFICATION 1 circumvents this infinity problem. Instead, a reasonable value around $\mathrm{E}\{\mathrm{Im}\{s[n]\}^2\} = 1$ is chosen and the comparison of the imaginary part eliminated "at the other end" by using the $-\mathrm{Re}\{y[n]\}$ expression. The hypothetical effect is that the SGD algorithm will adjust its output $y[n]$ more likely towards a real-valued BPSK because the imaginary part of the error will always be higher (because of the too-high-chosen $\mathrm{E}\{\mathrm{Im}\{s[n]\}^2\} = 1$) than the real part. This way, the algorithm will address any "illegal" rotations in the complex plane towards the imaginary axis *first*. The powerful yet simple error has the final form

$$\boxed{e_{mod}[n] = \Big(\frac{\mathrm{Re}\{y[n]\}^3}{1} + j\frac{\mathrm{Im}\{y[n]\}^3}{1} - \mathrm{Re}\{y[n]\}\Big)} \tag{3.38}$$

Note that MODIFICATION 1 can also be applied to the CMA (Godard) error function if it is re-arranged slightly. Simulations show the same effect as in the "W_{new}" case, thus once more supporting the hypothesis (independent of a specific algorithm). From (3.25) with $p = 2$ and BPSK input (expectation-term = 1), the CMA reformulates as follows:

$$\begin{aligned} e_{cma}[n] &= \Big(|y[n]|^2 - 1\Big) \cdot y[n] \\ &= y[n] \cdot |y[n]|^2 - y[n] \qquad \overset{mod}{\Rightarrow} \qquad \boxed{y[n] \cdot |y[n]|^2 - \mathrm{Re}\{y[n]\}} \end{aligned} \tag{3.39}$$

In this form, CMA bears remarkable resemblence to the W_{new} error. They both are associable with a general "power of 3" on $y[n]$. However, the CMA has a different impact on the imaginary part of $y[n]$ because of the absolute value – whereas in W_{new} the real and imaginary part are completely separated.

Before the algorithm can be stated in its final version, a second important modification is introduced. This modification has already been established in literature. It is called *normalization* and it is well known in the context of the LMS algorithm. Haykin [26] describes the normalization of LMS as a stabilizer regarding the coefficient change of the equalizer. This procedure minimizes the squared Euclidian norm of the change $\Delta\boldsymbol{c}[n+1] = \boldsymbol{c}[n+1] - \boldsymbol{c}[n]$. The effect of this is explained to the point by the "principle of minimal disturbance" (Widrow/Lehr, 1990), which states [26, p.436] that *in the light of new input data, the parameters of the adaptive system should only be disturbed in a minimal fashion.* Normalization makes the stepsize parameter independent of the tap-input power. Otherwise, the input power variation must be studied and the stepsize chosen accordingly to guarantee convergence. With normalization, stepsizes of $0 < \mu < 2$ lead to a convergent algorithm. It has also been observed that the normalized (LMS) algorithm converges potentially faster than its unnormalized variant. Following LMS update equation (3.) in Table 3.1, the normalization is inserted as a division of the stepsize by the squared Euclidian norm of the tap-input vector $\boldsymbol{r}[n]$:

$$\boldsymbol{c}[n+1] = \boldsymbol{c}[n] + \frac{\mu}{||\boldsymbol{r}[n]||^2} \cdot \boldsymbol{r}[n] \cdot e^*[n] \tag{3.40}$$

In the implementation of normalization a small constant is usually added in the denominator to avoid division by values close to zero (if $||\boldsymbol{r}[n]||^2$ is very small at any instant n).

MODIFICATION 2

Choose $\alpha > 0$ but sufficiently small in order not to bias the norm. Choose $0 < \mu < 2$, where $\mu = 1$ theoretically yields optimal convergence speed. However, it is recommended to choose $\mu < 1$ in this context.

$$\boldsymbol{c}[n+1] = \boldsymbol{c}[n] + \frac{\mu}{\alpha + ||\boldsymbol{r}[n]||^2} \cdot \boldsymbol{r}[n] \cdot e^*[n] \tag{3.41}$$

Although this modification has been introduced in the context of LMS, it has been shown in [41] that common standard adaptive equalizers such as LMS, NLMS, RLS or even the projection algorithm may be modified to become blind equalizers while keeping the least-squares-specific advantages, respectively. This is done by invoking a so-called "separation principle". A remark

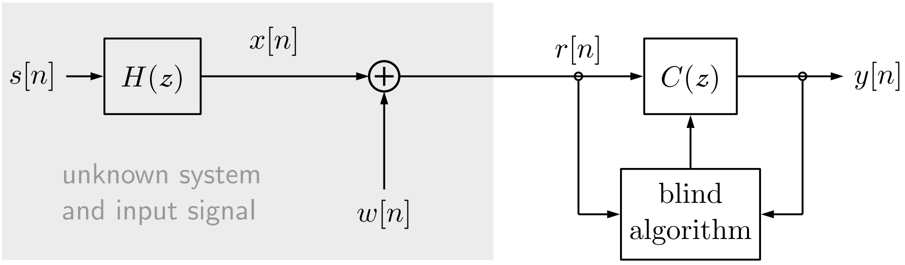


Figure 3.7: Signal flow graph of blind adaptive equalization

wNCMA algorithm	
0. Initialize $\boldsymbol{c}[n]$	
1. Filter output:	$y[n] = \boldsymbol{c}^H[n]\boldsymbol{r}[n]$
2. Current error:	$e_{mod}[n] = \mathrm{Re}\{y[n]\}^3 + j\mathrm{Im}\{y[n]\}^3 - \mathrm{Re}\{y[n]\}$
3. Coefficient update:	$\boldsymbol{c}[n+1] = \boldsymbol{c}[n] + \frac{\mu}{\alpha+\|\|\boldsymbol{r}[n]\|\|^2} \cdot \boldsymbol{r}[n] \cdot e^*[n]$

Table 3.2: wNCMA algorithm procedure and equations

at this point: The general idea is likely to hold given all of these algorithms belong to the SGD family. The convergence process of blind equalizers may, however, be more "turbulent" as opposed to LMS, thus the recommendation for a $\mu < 1$.

The choice of a name for this algorithm, i.e., wNCMA, may be a bit controversial. Usually, authors append a combination of new letters to an existing abbreviation to signify their modification (e.g., MCMA, RD-CMA, FSE-CMA). The name of the underlying algorithm "W_{new}" lacks referential meaning. For this reason, the three letters CMA are chosen instead to signify that (1.) the underlying algorithm itself fulfils the constant modulus criterion and that (2.) it is closely related to the original CMA. The capital "N" suggests normalization (see Modification 2) and the lower-case "w" in front hints at the true origin (W_{new}) as well as its well-behaved character concerning the number-set issue (see Modification 1).

4

Results & Evaluation

This chapter presents simulation results involving the new algorithm introduced in Section 3.5. Section 4.1 describes the general settings for the simulations. Section 4.2 details equalizer parameter choices and initialization. The bit error rate is used as a standardized measure for equalizer performance, and results are presented in Section 4.3. In Section 4.4 the validity of MODIFICATION 1 (see Section 3.5) is experimentally proved and discussed. Finally, Section 4.5 covers the case of a synchronization period in front of the message signal.

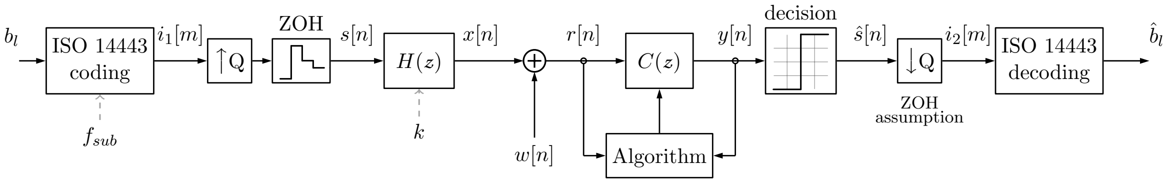


Figure 4.1: Signal flow graph of simulation setup

4.1 Simulation setup

The simulation has been designed with a practical orientation. The fundamental premise is the constant sampling frequency of $f_s = 13.56\,\text{MHz}$ in the receiving device. Since the ISO-14443 norm [5] defines several subcarrier frequencies along with corresponding baud rates (see Table 1.1), the simulation becomes implicitly *multirate*. This fact is indicated by the two boxes in Figure 4.1 labeled with "Q", which is an integer upsampling (resp. downsampling) factor depending on the choice of subcarrier (f_{sub}). Q is set so that the intermediate signal

Signal	Sample rate	Domain
b_l	none (binary sequence)	$\{0, 1\}$
$i_1[m]$	chosen f_{sub}	$\{\pm 1\} \in \mathbb{R}$
$s[n]$	13.56 MHz	$\{\pm 1\} \in \mathbb{R}$
$H(z)$	13.56 MHz	$\mathbb{C}$
$x[n]$	13.56 MHz	$\mathbb{C}$
$w[n]$	13.56 MHz	$\mathbb{C}$
$r[n]$	13.56 MHz	$\mathbb{C}$
$C(z)$	13.56 MHz	$\mathbb{C}$
$y[n]$	13.56 MHz	$\mathbb{C}$, but $\approx \mathbb{R}$
$\hat{s}[n]$	13.56 MHz	$\{\pm 1\} \in \mathbb{R}$
$i_2[m]$	chosen f_{sub}	$\{\pm 1\} \in \mathbb{R}$
$\hat{b}_l$	none (binary sequence)	$\{0, 1\}$

Table 4.1: Simulation sample rates of each intermediate signal

$i_1[m]$ is converted to an equivalent 13.56 MHz signal ($s[n]$) using zero-order hold (ZOH). After the equalizer output ($y[n]$) the reversed process is applied in order to retrieve ISO-conform signals. The decision device is assumed to impose rectangular form on the output signal again, so that the signal $\hat{s}[n]$ may be decimated without additional filtering. It must be pointed out that this receiver structure is a very primitive one. Advanced methods do exist (e.g., matched filtering, Viterbi decoding, see outlook in Section 5.2), however, the *channel equalization* is under examination here, not the posterior decoding structure. Using a basic decoding structure bears clear results regarding the equalizer.

Table 4.1 shows the sample rates for each intermediate signal (or system), and also their underlying number sets. Apart from the appended coding blocks in Figure 4.1, the center of the signal flow concept is identical to the one presented in the previous chapter in Figure 3.7. The lower box is intentionally labeled “Algorithm” to signify that not only the proposed blind algorithm is simulated, but also other blind algorithms as well as MMSE for comparison. In the following, each stage in the signal flow is described:

1. The binary messages b_l are generated randomly.
2. A subcarrier frequency is selected (0.848, 1.695, 3.39 or 6.78 MHz, comp. Figure 1.2).
3. b_l is modulated onto the subcarrier according to ISO-14443 (see figure 2.3, top three subplots). This intermediate signal is called $i_1[m]$.
4. The signal resulting from (3.) is upsampled to $f_s = 13.56$ MHz using zero-order hold. The factor “Q” is 1 for the highest $f_{sub} = 6.78$ MHz (no upsampling), Q = 2 for 3.39 MHz, Q = 4 for 1.695 MHz, and Q = 8 for 0.848 MHz. The upsampled signal is called $s[n]$.
5. Parameter coupling is selected $k \in [0.01 \;\; 0.55]$.
6. The baseband system response $H(z)$ for the selected k is provided by the system model (see chapter 2) as IIR filter.
7. $s[n]$ is filtered by $H(z)$ yielding $x[n]$.
8. Parameter signal-to-noise ratio (SNR) is selected and normalized by an approximation of the energy of the channel response to have a measure E_b/N_0.
9. Complex Gaussian noise $w[n]$ (AWGN) is added to $x[n]$ according to E_b/N_0. The resulting signal is $r[n]$.
10. In order to simulate automatic gain control (AGC), $r[n]$ is normalized by its RMS.
11. $r[n]$ is filtered by the current equalizer response $C(z)$, which is a transversal FIR filter of length L. $C(z)$ is iteratively adapted by an “Algorithm” of choice, which employs an error signal $e[n]$. The result of the filter operation is $y[n]$.
12. The decision device (slicer) is implemented as a simple signum function: $\hat{s}[n] = sgn(y[n])$. The signum function is defined with regard to complex numbers.
13. The recovered signal $\hat{s}[n]$ is assumed to have rectangular shape (like $s[n]$). With respect to the zero-order hold upsampling in (4.), the signal is downsampled to subcarrier rate again (using the same factor “Q” as in (4.)) by keeping the first of each q samples. The intermediate result is called $i_2[m]$.
14. The binary message $\hat{b}_l$ is reconstructed by demodulating the BPSK signal $i_2[m]$ from the subcarrier.

The simulation is implemented in MATLAB. Specific parameter settings are stated in the respective sections where their variation is studied or their preference explained. It is important to mention that the synchronization frame specified in ISO-14443 is explicitly not part of the following simulations. The synchonization frame is treated separately in Section 4.5.

4.2 Equalizer settings and initialization

The following sections comment on the choice of parameter settings for the blind equalizer. They apply to wNCMA as well as the standard NCMA; parameters are identical to provide a fair comparison in Section 4.3. They also apply to all algorithm variants considered in Section 4.4, since these variants differ in their error definition $e[n]$ only.

4.2.1 Stepsize μ

For conventional LMS (see Section 3.2.3) the choice of stepsize μ is well known to be dependent on the eigenvalues of the autocorrelation matrix $\boldsymbol{R}$ of the received signal. This effect is alleviated by normalization (see MODIFICATION 2) of the received signal. The choice of μ in context of LMS is comparatively easy, because the cost surface is unimodal and thus has a fixed curvature everywhere. Blind equalizers usually have multimodal cost surfaces, which means the issues discussed in Section 3.3.3 apply – and outweigh – the influence of the stepsize μ. In an exemplary situation in [13] a CMA equalizer passes a saddle point on the surface during convergence: Regardless of the chosen stepsize μ, convergence is slowed down arbitrarily. The convergence rate for blind equalizers is thus not adjustable by a single parameter μ, but is rather a careful joint selection of μ and the initial coefficient setting (see Section 4.2.4). In vicinity of a global minimum on the blind equalizer's cost surface, the choice of μ is, however, closely related to the LMS strategy. Considerations about eigenvalues and normalization are valid for blind equalizers under these circumstances.

Normalization of the stepsize (MODIFICATION 2) is important with regard to coupling factor k. Without normalization, the upper bound for a stable stepsize is defined by the largest eigenvalue λ_{max} of $\boldsymbol{R}$: $0 < \mu < 2/\lambda_{max}$. This eigenvalue directly depends on the shape of the resonance at each given k, because the power spectrum (PSD) – the Fourier transform of the autocorrelation – will also exhibit this resonance shape. The maximum eigenvalue is directly related to the maximum of the resonant peak by $\lambda_{max} \leq max\big(P_{xx}(f)\big)$ (further details in the solution manual associated to [42]). In other words, μ depends on k. In order to guarantee the stability of the (unnormalized) algorithm for *all* k, μ would have to be chosen so small as to guarantee stability for critical coupling (because the resonance peak is at a maximum here). However, for all other k the stepsize would be suboptimal. Normalization reduces this dependence.

The stepsize μ also affects the steady-state behaviour of the algorithm. In LMS it produces excess error due to remaining fluctuations around the global minimum. Blind algorithms with multimodal cost surface will behave similarly near a minimum, however, additionally there is the chance of diverging again from a minimum if the stepsize is too large – especially with high noise levels when the surface is flattened.

Consequently, the stepsize is set to $\mu = 0.01$.

4.2.2 Equalizer length L

As stated in Chapter 2 the system model has two poles and two zeros. According to system theory, the two poles can be perfectly equalized using an FIR filter containing the corresponding two zeros that cancel these poles. The model zeros cannot be perfectly equalized by an FIR filter. But an approximation is possible. Fortunately, the poles of the channel model are of greater overall significance compared to the zeros. An equalizer with many coefficients L can thus be avoided by neglecting the cancellation of the model zeros.

When noise is present it is advantageous to have spare coefficients, because the equalizer can use them to compensate for some of the noise. Experiments to compute inverses of the given system model have been shown in Section 3.2.3. In Figure 3.5 the zeros of these inverses are plotted. Tests in this context have revealed that the optimal choice for the number of filter zeros is 4, which translates to an

$$\boxed{\text{Equalizer with } L = 5 \text{ coefficients.}}$$

4.2.3 Normalization bias constant α

This parameter is a small constant to avert division by zero during the normalization step (MODIFICATION 2). It depends on the number format used in the calculation. In the simulations for this thesis, double-precision floating point native to MATLAB is used. The machine epsilon is used as the bias constant.

$$\boxed{\alpha = \texttt{eps('double')} \approx 2.22 \cdot 10^{-16}}$$

4.2.4 Initialization

This section introduces guidelines how to set the initial coefficients $\boldsymbol{c}[n]$ of the blind equalizer. Many complications arising in the course of this thesis (e.g., Sections 3.3.3 and 4.2.1) suggest a diligent choice in this matter. Several demands have to be met for properly functioning blind equalizers in general – and in accordance to the given RFID system in particular. The desired initialization shall be able to

1. avoid regions of attraction attributed to undesired local extrema or saddles,
2. start off being already close to the correct global minimum,
3. minimize the transient period (convergence time),
4. converge to the correct global minimum regardless of current coupling k, and
5. converge to the correct global minimum regardless of noise level.

These requirements cannot be all fulfilled. The biggest challenge is the combination of the first with the forth demand: The given range of possible k-values permits a significantly changing cost surface. Extrema and saddles change dynamically with k. Finding a single initialization fit for *all* k has proved unsuccessful. However, a slight reduction of k's range (lowering requirements for $k \ggg$) achieves this univeral “single-fit” initialization (comp. figures in Section 4.3.2).

The natural approach is to start with using the results pre-computed by the LMS algorithm in Section 3.2.3. The best possible approximations of the model system's inverses for all k have been presented there (noiseless[27] situation; see Figure 3.5). These inverses have been applied systematically as the initial coefficients $\boldsymbol{c}_0$ of the blind equalizer (wNCMA) and their effect evaluated by simulation. The inverses related to very low k show successful convergence for channels up to $k \approx 0.15$. This may be attributed to the common single resonance of the system – or respectively single notch of the equalizer – at $f = 0\,\text{Hz}$ in the spectrum. Recall Figure 2.18 on page 33: The model spectra are plotted here. To imagine the inverse spectra,

---

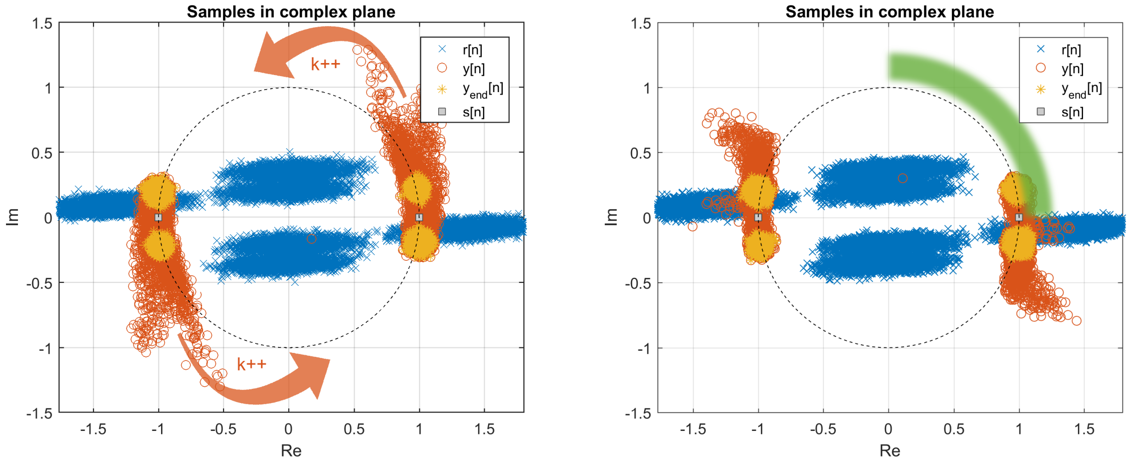
[27] i.e., the pre-computed inverses effectively correspond to an approximated zero-forcing (ZF) solution

turn the figure upside down. Evidently, the inverses at low k introduce very high gain to the filtered output (trying to make up for the weak coupling gain of the channel). This renders them unfit for initialization, because if the channel is, for instance, at a higher k of 0.3 (already providing sufficient gain), the starting equalizer may render the output signal unstable.

Initializing on the other end of the k-range has also been found to be problematic: The inverses in this region are spectrally quite flat and have two distinct notches (canceling the resonances). Simulations have shown that such initialization either tends to preserve two arbitrary notches or to diverge to an undesired minimum because the flatness of the initial spectrum of the equalizer does not clearly indicate where the blind equalizer is supposed to converge to. The single notch solution needed for the lower-k channels is rarely achieved by initializations with higher-k inverses.

The optimal choice is found in the middle of the k-range, more specifically in the vicinity of the point of critical coupling $k \approx 0.14$. At this point, the gain is maximal so the inverse initialization will never risk destabilizing the output. Moreover, the spectrum exhibits a single notch – although the zeros producing this notch are already clearly separable. This choice of initialization proved to achieve convergence towards both lower-k and higher-k systems (best trade-off). Requirement bullet number two is accommodated best by this choice (considering all k).

The last remaining challenge is the minimization of the transient period (3^{rd} requirement). Obviously, by initializing the equalizer with an already optimal or close-to-optimal setting the transient period will be eliminated. This is the case around $k = 0.14$. For all other k the equalizer will take a (short) time to converge. This is where the “over-modelling” of the complex domain becomes beneficial: The idea is to shift the convergence activity across the imaginary part of the output, so that the decision device can correctly clip the equalizer output to ± 1 *right from the start* for as many different k as possible. This can be achieved by a simple initial phase rotation appended to the initialization. The reasoning behind this is as follows: With k increasing from 0.01 upwards, the channel causes an increasing counter-clockwise rotation in the complex plane (the reason for this being the increasing phase distortions of the channel). Figure 4.2(a) illustrates this tendency. The values ($\in \mathbb{C}$) of signal samples are plotted



(a) wNCMA without rotation correction

(b) wNCMA with a 68.75° clockwise rotation

Figure 4.2: Initial rotation study in complex plane; low $k = 0.05$, $f_{sub} = 6.78$ *MHz,* 40 *dB SNR*

at different stages of the signal flow (see Figure 4.1). Signal $y[n]$ (red circles) is of interest here. The outlying circles mark the start of the equalizer (transient phase) and eventually congregate close to ±1 (yellow).

With increasing k (pale red arrow and `k++` to indicate/imagine an animated series of similar plots for increasing k) those outlying samples move counter-clockwise around the circle. The speed of that movement depends on the chosen subcarrier: Higher frequency subcarriers show greater movement. If the imaginary axis is crossed eventually,[28] convergence errors – and subsequently decision errors – are inevitable. In order to shift (to defer) the crossing event to the highest possible k value, a fixed clockwise rotation is appended to the initialization of the coefficients. This rotation can be seen as a manual compensation for the system's allpass part discussed in Section 3.1. Depending on the subcarrier, this rotation measures between 60 and 90°. Figure 4.2(b) shows the effect of a 68.75° rotation. The outlying red samples now congregate (approach the BPSK points) from the other direction. A spare range of approximately 90° (counter-clockwise) is available as a buffer (green) before the channel-induced rotation reaches the critical crossing region. The result is a stable operation up to approximately $k = 0.4$. Without the rotation, wrong decisions may occur as early as $k = 0.2$ (worst case, depending on the subcarrier).

The resulting overall delay of the combined channel plus equalizer system is improved by the rotation as well. Johnson [13] showed that blind equalizers tend to converge to a solution which possesses the same group delay as the initial setting. This means that the choice of initialization determines the delay of the overall system; convergence to different resulting delay times is most unlikely. Nevertheless, the channel exhibits a gradually changing delay when approaching greater k values. This delay is, however, shorter than the duration of 1 sample, which means it manifests itself as a fractional delay – which in this case is just another expression for the phase rotation discussed above. Associating the delay context to Figure 4.2, the crossing of the imaginary axis signifies that overstepping samples are linked to too great a delay time and thus cause wrong decisions. The compensating rotation (green) ensures that the major part of the k-range remains associated with a fixed chosen delay of 1 sample. Moreover, the compensation leads to a maximally short transient period *should* the delay switch to 2 samples yet (which is possible at high subcarrier frequencies and very high k, comp. Figures 4.6, 4.7 and 4.8(d): inverted signal, BER approaches 10^0).

The initialization strategy does not lose its legitimacy in the presence of noise. The notch shape of the equalizer also attenuates the noise spectrum in the notch frequency region. Apart from the notch, the spectrum is rather flat, so frequency-selective noise amplification does not occur. A constant noise amplification may occur due to the initial equalizer gain. However, such a constant noise gain has negligible effect on the overall convergence.

The selected initial coefficients (corresponding to the inverse model of $k = 0.14$) are

$$\boxed{\boldsymbol{c}_0 \quad = \quad \beta \quad \cdot \quad \begin{bmatrix} -0.425+j0.6 \\ 0.4-j0.556 \\ -0.0472+j0.0054 \\ -0.0149+j0.0127 \\ 0.0107+j0.0178 \end{bmatrix}}$$

f_{sub} (MHz)	β
0.848	$6e^{j1.2}$
1.695	$3e^{j1.2}$
3.390	$1.5e^{j\pi/2}$
6.780	$1.5e^{j\pi/2}$

Table 4.2: Subcarrier-dependent values for equalizer initialization

---

[28] around the arrow's head, when the first red circles would come to lie here

4.3 BER: Bit error rate evaluation

The bit error rate simulations are based on the settings stated in the previous Sections 4.1 and 4.2. Bit error rates are widely used to evaluate the performance of communication systems. Such a system is built of a communications channel (here: RFID channel plus AWGN) cascaded by an equalizer structure (here: wNCMA and others). Performance is measured by counting the errors that the overall system produces.

The bit error probability is approximated by the Monte-Carlo method: The simulation generates a random message signal $s[n]$ and calculates the results for the currently fixed parameter selection. This procedure is repeated for $M = 100$ times, afterwards results are averaged to yield a Monte-Carlo approximation. This approximation becomes more accurate for large sample sets N. The values of N vary[29] and are thus stated in the respective figure descriptions.

The signals used to calculate the bit error are the intermediate signals $i_1[m]$ and $i_2[m]$ (see Table 4.1). There are two reasons for this choice:

1. The equalizer's task is to restore the signal at subcarrier rate; the multirate setup (see Section 4.1) introduces a certain ambiguity if the errors were to be calculated at the rate of 13.56 MHz (especially for lower subcarriers), thus $i_2[m]$ is subject to the strict downsampling of Rule (13.) given in Section 4.1.
2. Usually the binary digits are used directly for BER calculation; in this case the ISO-14443 endorses a "nested" modulation scheme, where the binary message is first modulated onto a subcarrier, which is subsequently modulated onto the carrier. Assuming binary b_l ($\hat{b}_l$) and subcarrier signal $i_1[m]$ ($i_2[m]$) are merely different representations of a common underlying information stream, this nested modulation is avoided[30] by considering exclusively the subcarrier signals $i_1[m]$ ($i_2[m]$).

The overall system delay is assumed to be 1 sample. This choice accommodates most cases in the range of k (0.01 up to ≈ 0.4) under the given simulation settings.

4.3.1 Parameter variation: noise

Figure 4.3 cohesively shows performance results of the wNCMA equalizer. Since the equalizer is optimized for low k (see Section 4.2.4), all four subfigures show similar decreasing curves for these low k. The shape of the curves resembles the theoretical BPSK BER curve. Assume a threshold below which the incidence of errors is acceptably low (e.g., BER $< 10^{-2}$, which means 1 error out of 100 samples, or 1%). The wNCMA achieves this BER threshold (for low k, and in all four cases) at SNR (E_b/N_0) values between 15 and 20 dB.

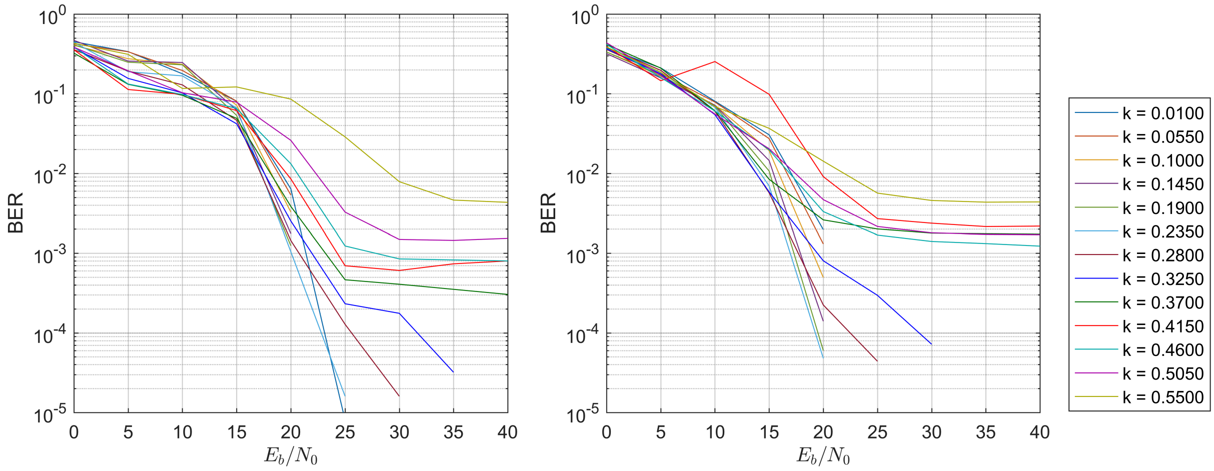
For curves $k > 0.3$ the BER does not decrese with increasing SNR. Instead, the curves approach a constant BER value (which remains the same even if SNR is increased to very high levels). The cause for this constant BER is the transient behavior of the wNCMA equalizer: Due to initialization with the "$k = 0.14$"-inverse, no or negligible transient errors occur at lower k. As k increases, the equalizer needs to converge from this (single-notch) initialization to a solution where the two notches are increasingly farther apart. Convergence time increases, and bit errors occur in the time span between equalizer startup and completed convergence. The Monte-Carlo average usually assumes the errors to be "evenly"[31] distributed. However,

---

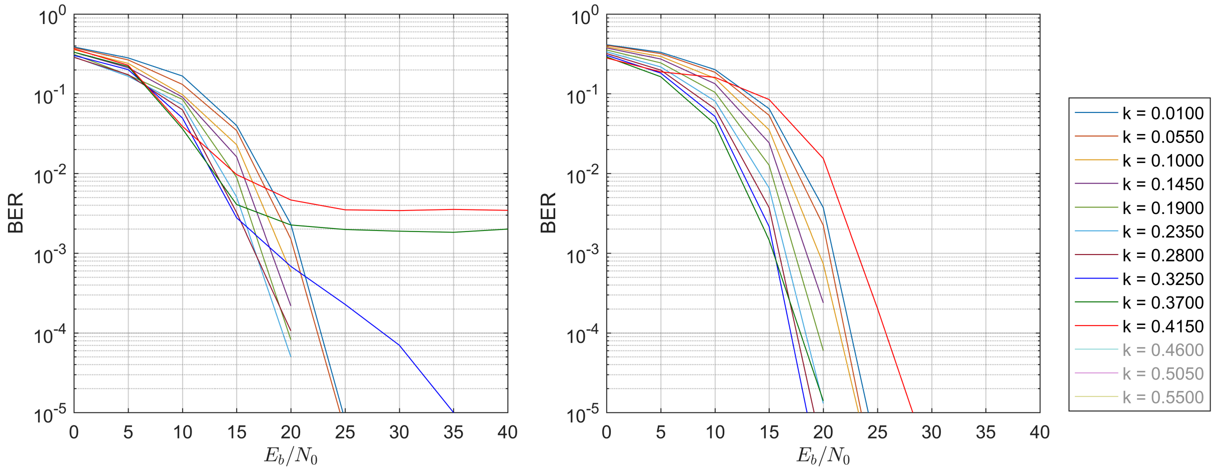
[29] due to runtime minimization of the simulation

[30] avoided so that the evaluation of the *equalizer* itself is not further distorted by ambiguous stages in the signal flow (same reasoning as for using simple, basic decoding methods)

[31] meaning spread across a certain range – as opposed to being localized in a certain part of that range



(a) Channel + wNCMA EQ → $f_{sub} = 0.848\,\text{MHz}$ (b) Channel + wNCMA EQ → $f_{sub} = 1.695\,\text{MHz}$



(c) Channel + wNCMA EQ → $f_{sub} = 3.39\,\text{MHz}$ (d) Channel + wNCMA EQ → $f_{sub} = 6.78\,\text{MHz}$

Figure 4.3: Rigorous BER parameter study of the proposed new wNCMA algorithm. $N = 10000$. *Main variation is* E_b/N_0. *Several* k *are plotted (right-side legends correspond to the pairs (a,b) and (c,d)). Each subcarrier rate is examined separately: (a)* 0.848 *MHz; (b)* 1.695 *MHz; (c)* 3.390 *MHz; and, (d)* 6.780 *MHz.*

here all the errors accumulating to the constant BER value are localized in the *beginning* of the simulated signal time. The first N_e of $N = 10000$ samples show errors – afterwards no errors occur,[32] just like in all the other cases for lower k.

The constant BER values (flat horizontal ends of lines in the Figures 4.3 (a), (b), and (c) at $E_b/N_0 > 25$) can be used to guess the approximate length of convergence. The constant part of the green curve in Subfigure (c) has a BER value of 0.002 ($2 \cdot 10^{-3}$). Multiplication by $N = 10000$ yields an absolute error total of 20. These errors are located in the *beginning* of the simulation time. Assuming these errors are slightly spread across the beginning phase, the convergence is likely to be over after $\approx 20 \cdot 5 = 100$ samples (generous estimate), the remaining $N - N_e = 9900$ samples are error-free.

---

[32] no errors due to the algorithm — random errors may still occur given low SNR levels

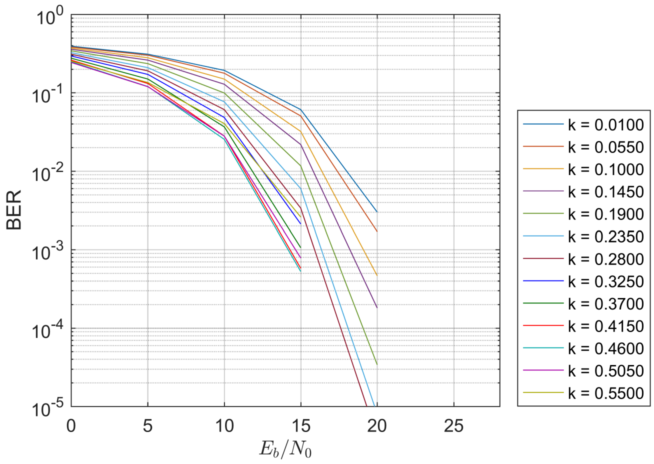


Figure 4.4: BER benchmark solution, $N = 10000$. *Tested system comprises the RFID channel plus MMSE inverse equalizer, with a subcarrier of* 6.780 *MHz. Other subcarrier cases look very similar. This figure is representative for all subcarrier rates. Acceptable performance is reached slightly before* 20 *dB* E_b/N_0 *for all* k.

When the transient errors are left out of consideration, the wNCMA equalizer achieves close-to-optimal performance similar to that of the MMSE equalizer in Figure 4.4. For subcarriers 0.848 MHz and 1.695 MHz the wNCMA achieves this performance level for *all* k; For the remaining two subcarriers, there is a limit of $k \approx 0.43$ up to which these performance levels are reached; beyond this limit the equalizer does not converge to the solution favoring the assumed overall delay of 1 sample. BER values of the grayed-out k curves in the legend (in Figure 4.3) would be constant around 0.25 to 0.5 (over the entire SNR range). Those curves have been omitted because they distort the intuitive information content of these figures – showing them would imply the wNCMA was unable to equalize at these k. This is, however, not true. The wNCMA equalizer does in fact equalize the message perfectly at such high k – but the overall signal delay is then 2 samples, and additionally the signal is inverted.[33] The question of whether this feature is of use in a practical implementation is deferred at this point.

4.3.2 Parameter variation: coupling

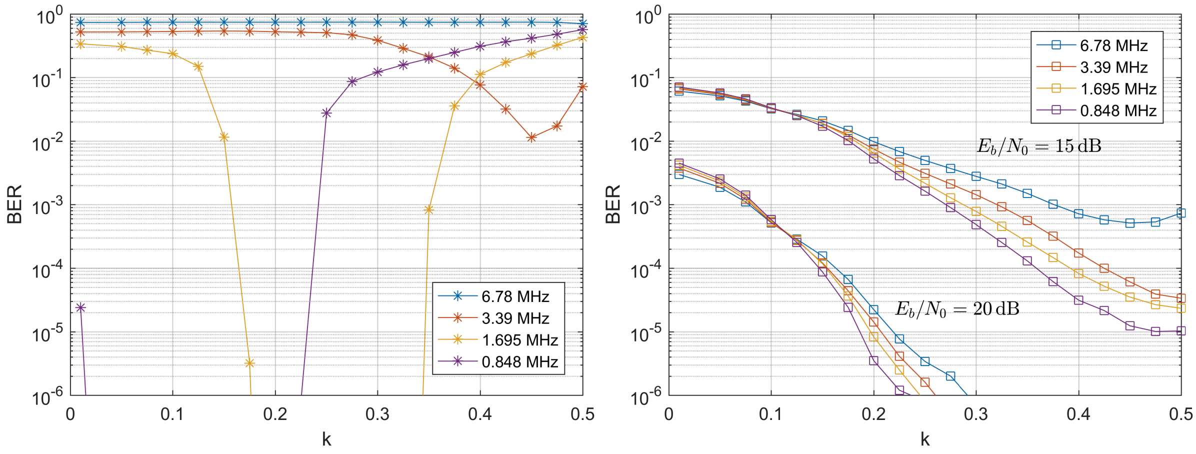
The critical SNR region for the wNCMA equalizer has been identified between 15 and 20 dB. In this section, the SNR is the fixed parameter. The BER is plotted against k in order to review the simulation results from a second perspective.

Caution is advised while studying Figure 4.5(a). The BER results of the unequalized RFID channel are subject to multiple influencing factors, which stem from model properties (phase distortion, allpass part) and from the chosen simulation setup (fixed delay of 1, multirate, simple decoding structure, downsampling with ZOH assumption). A comprehensive explanation is given in the following paragraph.

Figure 4.5(a) shows the bit errors in the case of *no* equalizer after the RFID channel. The violet curve indicates that the RFID system structure (comp. Figure 2.2 on page 15) has been tuned to favour the 0.848 MHz subcarrier. This is understandable, since all data rates in the ISO-14443 standard *lower* than 0.848 Mbit/s still use the 0.848 MHz subcarrier (see Table 1.1 on page 8). The reason for the low BER at low k for 0.848 MHz is a longer symbol duration due to oversampling (multirate, see Section 4.1) in conjunction with the chosen simple

---

[33] $(-1) \cdot s[n] \quad \equiv 180^\circ$ phase inversion



(a) unequalized channel BER → $E_b/N_0 = 15\,\text{dB}$

(b) MMSE equalizer benchmark

Figure 4.5: $N = 100000$. *Opposites: (a) unequalized channel errors drop occasionally due to multiple factors; (b) optimal MMSE equalizer achieving good performance over all* k *when approaching* 20 *dB SNR* (E_b/N_0)*, pre-computed [offline], uses training sequences.*

decoding structure (see Figure 4.1): One subcarrier period equals 16 samples. Even if ISI is severe (low k), the interference is allowed to "smear" signal energy over each subsequent 8 samples before it would start degrading the information content.[34] For 1.695 MHz (yellow curve in Figure 4.5(a)) the permitted interference width is already reduced to 4 samples, respectively. This causes errors at lower k due to ISI. This lower limit is shifted towards higher k as the oversampling factor (Q) decreases (see red curve in Figure 4.5(a); the blue curve is fully affected by ISI because there is no oversampling at 6.78 MHz).

Moving on to the discussion of high BER values for $k > 0.25$ (0.848 MHz) and $k > 0.35$ (1.695 MHz), the phase rotation of the RFID channel comes into play. These phase distortions are attributed to the allpass system part (see Section 3.1). Their strength of impact varies depending on f_{sub} and k. For 0.848 MHz they shift the signal out of the valid "overall 1 sample delay"-region at $k = 0.25$ already; for 1.695 MHz the same happens at $k = 0.35$ and so on (for 3.390 MHz it is beyond the right plot border).

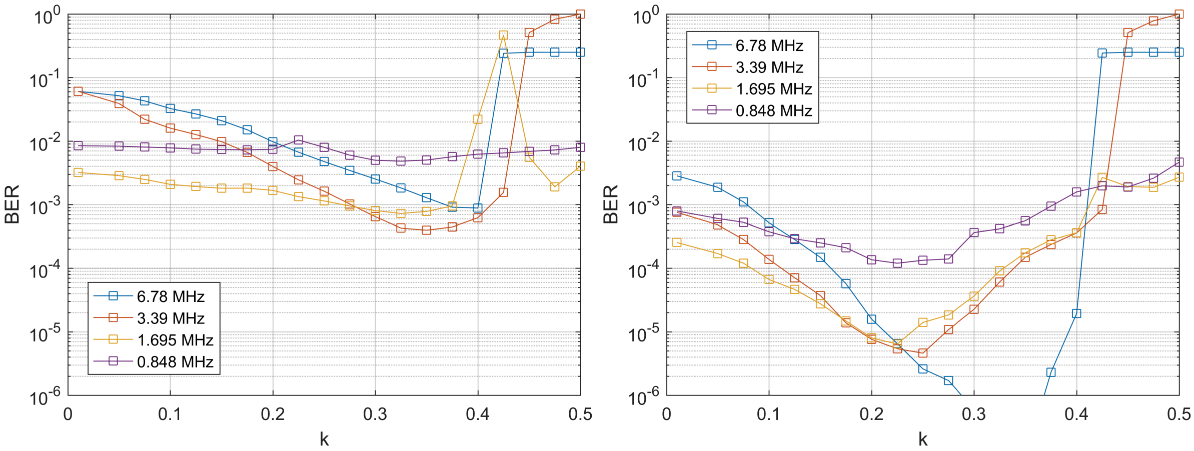
The neighboring figure 4.5(b) shows the achievable[35] optimum of equalization. Two bundles of curves are plotted, for 15 and 20 dB SNR. The area between the bundles is the critical performance-limiting region. For 15 dB SNR the BER is still high especially at lower k. With increasing SNR towards 20 dB, the BER will drop below the 10^{-2} threshold – and show a steeper decrease towards higher k values.

Following the example of Figure 4.5(b), the results for the wNCMA equalizer are plotted in the same manner. Since here the curve bundles are not as well-behaved as the optimal MMSE curves, the two presented SNR cases are split into two plots. The general intention of the four plots on the next page is to show the critical SNR region of wNCMA (Figure 4.6(a) and (b)) and to subsequently compare the performance of wNCMA to the standard NCMA (Figure 4.7(a)) and the standard NCMA with MODIFICATION 1 (Figure 4.7(b)) for the common fixed E_b/N_0 of 20 dB.

---

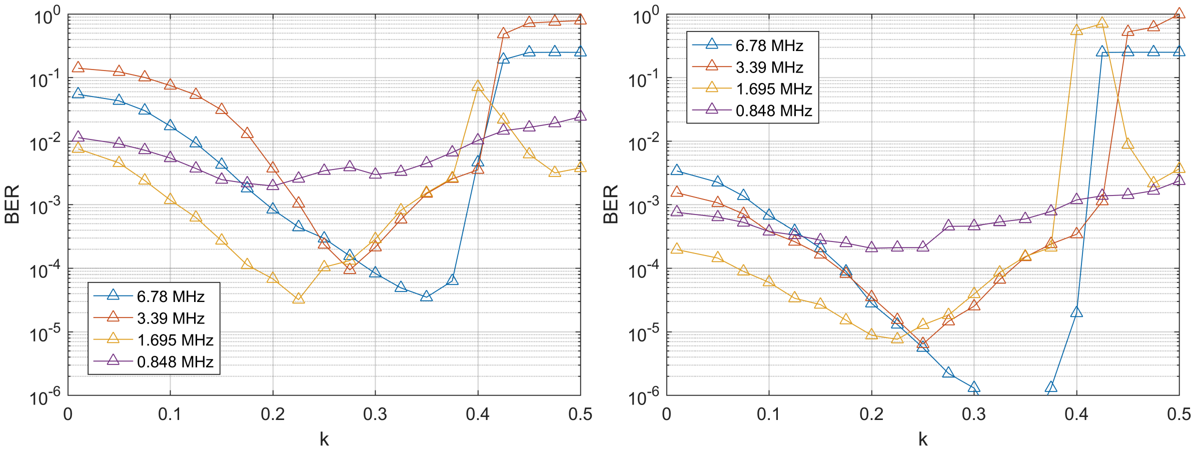
[34] because each *first* sample out of these 8 is kept after downsampling; comp. Rule (13.) in Section 4.1

[35] achievable in theory/simulations – not realizable in RFID context because of required training sequences.



(a) wNCMA BER → $E_b/N_0 = 15\,\mathrm{dB}$ (b) wNCMA BER → $E_b/N_0 = 20\,\mathrm{dB}$

Figure 4.6: $N = 100000$. *BER of the equalized RFID channel: Transition region from acceptable to good performance (SNR-wise) of the wNCMA equalizer.*



(a) standard NCMA BER → $E_b/N_0 = 20\,\mathrm{dB}$ (b) NCMA + Modific.1 BER → $E_b/N_0 = 20\,\mathrm{dB}$

Figure 4.7: $N = 100000$. *BER of comparable equalizers: (a) common CMA (normalized); (b) CMA (normalized) and the additional* MODIFICATION 1 *to favor real-valued BPSK output. Both for* 20 dB *SNR* (E_b/N_0).

The eye-catching feature of all four subfigures is the jump to high error rates at $k > 0.4$. The previous discussion about the unequalized channel (Figure 4.5(a)) applies here, more specifically, the part about the phase rotation. The difference now is that the wNCMA (and the other two algorithms) are capable of redefining the point where the delay shift from 1 to 2 samples occurs. The delay shift happens around $k = 0.43$, which is a result of equalizer initialization.

For low k and acceptable SNR levels (see Figure 4.6(b)), the wNCMA equalizer performs very close to optimal (comp. Figure 4.5(b)). However, for critial SNR levels the influence of AWGN may paralyze the equalizer (see flat violet curve in Figure 4.6(a)). In the region of $0.3 < k < 0.4$, the BER values increase again (all four plots) due to the already explained transient convergence period which – being simulated in the critical SNR region – is additionally deteriorated by the increased noise levels.[36]

---

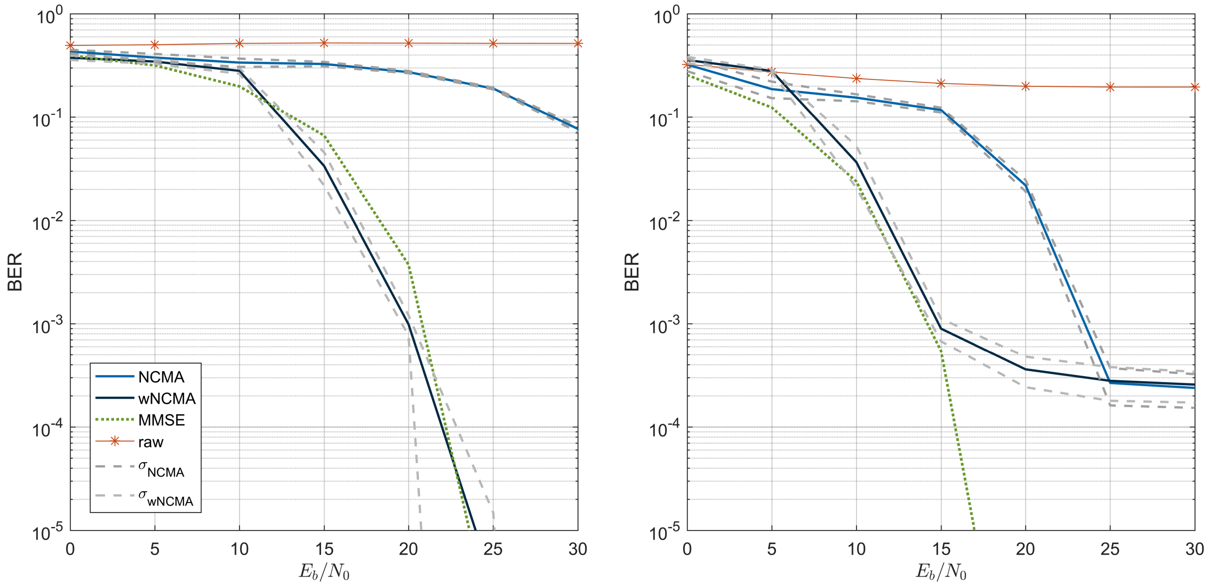
[36] equalizer misadjustment + random noise = increased likelihood of wrong symbol decisions

Now the differences between wNCMA (Figure 4.6(b)), standard NCMA (Figure 4.7(a)) and standard NCMA with MODIFICATION 1 (Figure 4.7(b)) will be discussed. Starting with the standard NCMA (Figure 4.7(a)), the error rate is significantly higher. The main reason are systematic errors due to the *number-set-dependent minima* which mislead the CMA to assume higher-level PSK constellations. The overall curve shapes remain similar to the other two compared equalizers, which can be traced back to the well chosen initialization (which is identical in all three cases). Enabling MODIFICATION 1 within the standard NCMA immediately reduces the bit error rate by a significant amount (see Figure 4.7(b)). The modified version achieves BER curves very similar to those of the wNCMA (see Figure 4.6(b)). The difference lies in the behavior of two algorithms (wNCMA and NCMA+MODIFICATION 1) in vicinity of the delay switch region ($k \approx 0.43$). The modified NCMA shows a wider critial zone with ambiguous convergence behavior, especially for 1.695 MHz (yellow curve). The wNCMA exhibits a more reliable convergence behavior in this k-region, which counts in favor of the wNCMA. Further details are presented in Section 4.4.

4.3.3 Summary: wNCMA comparison to NCMA and MMSE

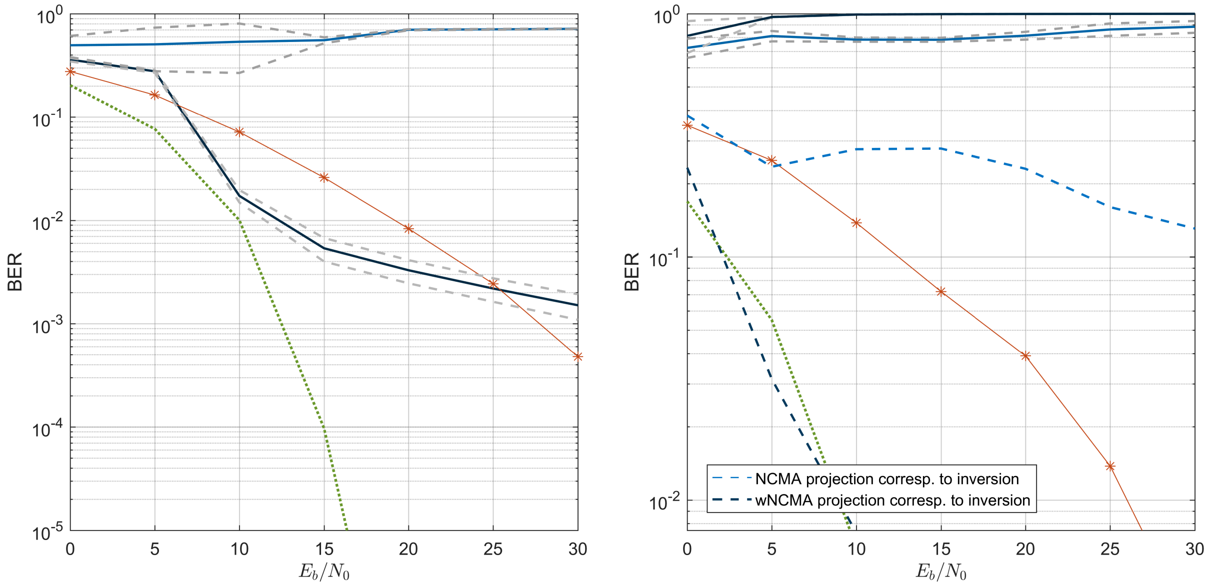
In order to give a concluding perspective on the performance evaluation, Figure 4.8 shows four subplots which directly compare the wNCMA equalizer to NCMA and MMSE. The four subplots have been chosen such that they represent the most important stages in the range of k. The common subcarrier of 3.39 MHz is chosen out of the four possible options for f_{sub}, because it most fittingly represents the other three f_{sub} parameter studies in the cases of Subfigures 4.8(a), 4.8(b) and 4.8(c). Additionally, wNCMA's ability to achieve a valid (but inverted) reconstructed message signal can be best emphasized for $f_{sub} = 3.39\,\text{MHz}$ in Subfigure 4.8(d).

Figure 4.8(a) shows the BER results for $k = 0.01$. It shows that wNCMA can achieve optimal MMSE performance for very low coupling k. Moreover, the general placement of the curves stays approximately the same for all k up to 0.3. The NCMA (light blue) always performs inferior to wNCMA (dark blue) in this range of k. As an example of the next stage, Figure 4.8(b) shows the remaining constant BER due to the transients inherent in both NCMA and wNCMA. Interestingly, their convergence time is similar. Figure 4.8(c) shows the wNCMA equalizer being able to converge in the critical case of $k = 0.43$ (delay turning point), while the NCMA already fails. Figure 4.8(d) zooms in on the region of high BER values. The wNCMA quickly approaches a BER value of 1. This means that *every* symbol is wrong, which followingly implies that *inverting* the signal that is used in the error computation will result in *no* symbol being wrong. The wNCMA equalizer is thus able to reconstruct an error-free message signal for $k > 0.43$ – with the little drawback that it is not precisely synchronized anymore, meaning either a signal inversion (as in the case of $f_{sub} = 3.39\,\text{MHz}$) or one additional sample of delay (delay = 2 as in the case of $f_{sub} = 6.78\,\text{MHz}$). Either way, wNCMA performs well over a wide range of k while obeying the simulated synchronized delay of 1 sample. Concluding the BER evaluation, the wNCMA equalizer appears to be the best option to counter the effects of the RFID ISI channel: It outperforms the standard NCMA (and also the modified standard NCMA), and the supervised adaptive MMSE equalizer is out of consideration because it requires training sequences. The leading role of wNCMA stems from MODIFICATION 1, which is evaluated by simulation in the next section.



(a) Common parameter $k = 0.01$

(b) Common parameter $k = 0.35$



(c) Common parameter $k = 0.43$

(d) Common parameter $k = 0.50$

Figure 4.8: Equalizer comparison. $N = 40000$, $f_{sub} = 3.39$ MHz. The legend in (a) is valid for all subfigures.
(a) The general alignment of the curves is representative of $0.01 < k < 0.3$; wNCMA outperforms NCMA and is close to the MMSE benchmark;
(b) The (now manifesting) transient convergence period causes bit errors persistent even in good E_b/N_0 conditions > 25 dB (both blind algorithms). When excluding the transient period, wNCMA generally performs close to the MMSE benchmark, while NCMA is still behind by ≈ 10 dB;
(c) Critical region, wNCMA eventually converges (transient errors), NCMA fails, the unequalized channel ('raw', red) performs fairly well in this region (due to simulation setup, comp. Figure 4.5(a) red curve);
*(d) Past the critical point, wNCMA can achieve a valid but **inverse** solution (approaching BER = 1; signal phase 180°). The projected result (when taking inversion into account) shows that wNCMA would perform well, while NCMA fails. Note: The y-axis is zoomed in here to magnify the BER curves approaching 1 (10^0).*

4.4 Study of Modification 1

The following simulations show the validity of Modification 1 (see Section 3.5). Modification 1 drives the convergence process towards a real-valued output signal, assuming a real-valued input passed through a complex-valued channel before. This verification is of experimental nature. The modification has been tested using the RFID channel model and two different blind algorithms – in order to show that the modification can be applied not only to the proposed new algorithm, but also to other common blind equalizers.

In the following, the classic normalized CMA is compared to the new wNCMA. Both equalizers are presented *with* and *without* the modification. The received signal $r[n]$ (blue) is identical in all four cases, as are initialization and parameters k, f_{sub}, SNR, and number of samples. The whole set of equalizer output samples $y[n]$ (red) is plotted; additionally the last 25% of these samples are plotted again (yellow) to emphasize where the respective algorithms converge to. The gray square ($s[n]$) indicates the ideal constellation points for BPSK input ($\pm 1{+}j0$). The critical region around $k = 0.43$ is chosen intentionally, because there the combination of channel and equalizer (initialization) leads to a short transient (adjustment) interval. Different directions of convergence of the contesting algorithms are emphasized this way. For lower k the differences are less distinct (even shorter or no transient interval).

Figures 4.9(a) and (b) show the change in convergence behaviour of NCMA due to Modification 1. The complex-valued channel affects the unmodified NCMA in Subplot (a) in such a way that it converges towards multiple points spread across the unit circle. The constant modulus criterion is fulfilled, however, the algorithm assumes a constellation shape resembling an incomplete and distorted 8-PSK rather than BPSK. Notice that the yellow points of convergence are not due to oversampling: $f_{sub} = 6.78\,\text{MHz}$ means two samples per symbol. Still, as many as four additional points appear on the circle (additional regarding those two in vicinity of the optimal BPSK squares). Consequently they have to be attributed to the complex-valued nature of the channel (in interaction with the equalizer).[37]

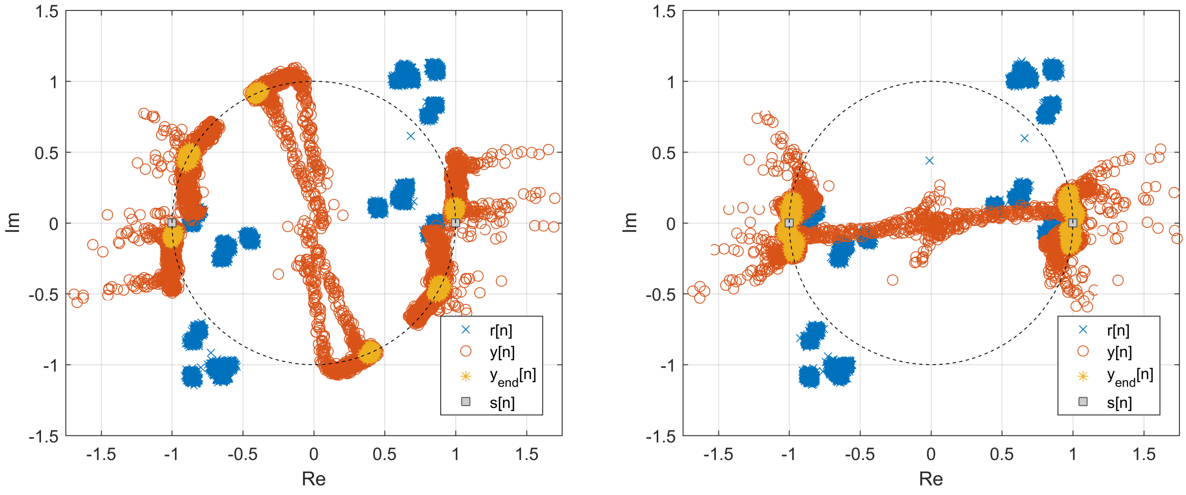
When Modification 1 is applied to NCMA in Figure 4.9(b), the algorithm does not converge to other points on the unit circle except those in close vicinity to the optimal BPSK squares. Notice that convergence starts at sample locations identical to the ones shown in (a). In (a), the initial samples are automatically drawn towards the nearest point on the circle. In (b), they are instead attracted to $\pm 1{+}j0$ regardless of where they started.

Figures 4.9(c) and (d) show the change in convergence behaviour of wNCMA due to Modification 1 analogous to the previous subfigures. The original W_{new} algorithm (c) converges to a QAM-like constellation – which is much more reasonable considering BPSK over a complex-valued channel. In other words, the W_{new} algorithm (in its unmodified form) is already able to retain *some* input properties. Additional plots (which cannot be presented here due to space restrictions) support this observation: They show convergence towards those two vertical lines intersecting the BPSK squares at $\pm 1{+}j0$. So in spite of the complex-valued channel effect a decision device would perfectly be able to decode a BPSK signal once the unmodified algorithm has converged. And exactly here lies the problem: Figure 4.9(c) shows an unnecessary prolongation the convergence time by resorting to suboptimal paths (first vertical, then horizontal). Furthermore, with decreasing subcarrier frequency the number of samples per symbol increases, which can cause the unmodified W_{new} to assume rather bizarre constellations due to oversampling.[38]

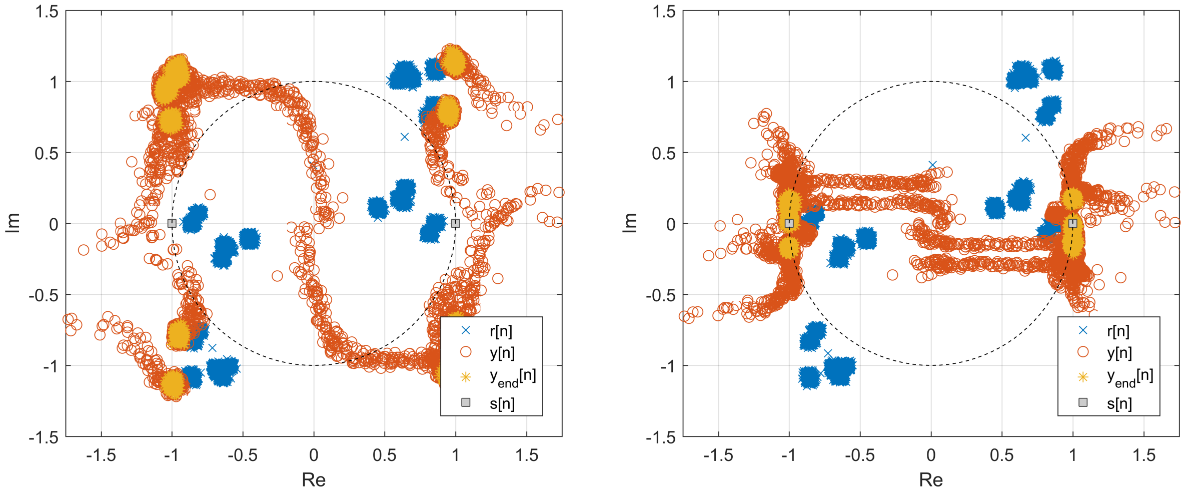
---

[37] Further evidence shows that the ISO-14443 conform input signal (Manchester coding in principle) also disqualifies as the cause of the additional constellation points. Simulations with random binary input instead show these additional points as well. Moreover, their positions are even more erratic in this random test case. Manchester coding appears to mitigate the erraticism to a certain extent (tendency).

[38] note that in the derivation of W_{new} no assumptions were made on the input constellation



(a) NCMA without Modification 1 (b) NCMA with Modification 1



(c) normalized W_{new} without Modification 1 (d) normalized W_{new} with Modification 1 = wNCMA

Figure 4.9: Modification 1 study. Sample values of various signals within the simulation flow graph (Figure 4.1) plotted in the complex plane. Common parameters: $N = 20000$ samples, $k = 0.43$, $f_{sub} = 6.78\,MHz$, $E_b/N_0 = 40\,dB$, same input signal, same equalizer initialisation (L, $\mathbf{c}_0$). Variation: disabled/enabled Modification 1 for NCMA in (a) and (b), and for W_{new} in (c) and (d), where the latter is the one called wNCMA.

When Modification 1 is applied to W_{new} – which is then given the name wNCMA – in Figure 4.9(d), a similar convergence behavior can be observed with respect to (b). The initial samples are also drawn to $\pm 1 + j0$. However, there is a difference in the alignment after convergence (yellow samples): In Figure 4.9(b) the tendency for a circular alignment is still given (due to the constant modulus criterion). In contrast, Figure 4.9(d) generally favors a straight vertical alignment.

An additional important observation is made when animating these figures over a gradually changing coupling factor k.[39] The unmodified NCMA is highly susceptible to phase rotations in the system. The visible yellow dots move counter-clockwise around the circle with increasing k. Once a yellow dot passes the imaginary axis, the NCMA will converge to the wrong solution or at least cause severe (regular) decision errors (as is the case in Figure 4.9(a),

---

[39] Since such animations (or image sequences) are impractical to be printed, a description has to suffice here.

where the diagonal dots at approx. +115° and −65° *have* crossed the imaginary axis already). When animating the modified NCMA over all k, a circular movement of the yellow dots can be observed, too. However, this time the movement occurs clockwise around small-radius circles with their center points being the BPSK points $\pm 1{+}j0$!

Analogous animations over all k with the unmodified W_{new} algorithm show erratic constellations. The unrestricted input assumption results in arbitrary constellations for each new k in the increasing sequence (e.g., imagine additional yellow points forming at $\pm j1$ in Figure 4.9(c)). In this state, the W_{new} algorithm is most unsuitable as an equalizer for the given channel. MODIFICATION 1 significantly improves the situation (compare Figure 4.9(d)). Arbitrary constellations are still possible (although much less frequent), only now they manifest themselves in very close vicinity to the BPSK points $\pm 1{+}j0$; like in the animated NCMA, those points are the center points – not of a circle, but of the arbitrary constellations. There is no additional circular movement around these center points like with the NCMA.

In summary, MODIFICATION 1 has been validated by simulation. Both modified versions of the compared algorithms show improved behaviour regarding the real-valued BPSK input over the complex-valued channel and equalizer. The preference of wNCMA over the variants presented here is demonstrated in Section 4.3 by means of bit error rates.

4.5 Synchronization time

The ISO-14443 standard [5] defines a synchronization period before the message signal starts. Synchronization is vital to avoid the inversion ambiguity of BPSK. It is also necessary for the implementation of more detailed transfer protocols (frames and block structures, see [3,4]). The relevant excerpt of the norm is quoted in the box below [5, p.32]. Note that PICC is the card, and PCD is the reader.

> - After any command from the PCD, a guard time TR0 shall apply in which the PICC shall not generate a subcarrier. TR0 shall be greater than 1024/fc (≈ 75,5 μs).
> - The PICC shall then generate a subcarrier with no phase transition for a synchronization time TR1. This establishes an initial subcarrier phase reference Ø0. TR1 shall be greater than 80/fs.
> - This initial phase state Ø0 of the subcarrier shall be defined as logic “1” so that the first phase transition represents a change from logic “1” to logic “0”.
> - Subsequently, the logic level is defined according to the initial phase of the subcarrier.
>
>   Ø0 represents logic “1”
>
> $Ø0 + 180°$ represents logic “0”

Figure 4.10 shows a simulated situation according to the ISO-14443 text. On the left, the steady subcarrier signal is evidently correctly synchronized, because the thick green line and the dashed black line are perfectly aligned here. The wNCMA equalizer has been activated at the start of TR1, it has had the time to adjust to the subcarrier signal. After 1200 samples, the synchronization is terminated by the first phase transition (vertical magenta line). Two details can be observed:

1. The amplitude of the equalizer output signal (red) suddenly increases.
2. Errors occur after the decision device: The thick green and the black dashed line are in mismatch.

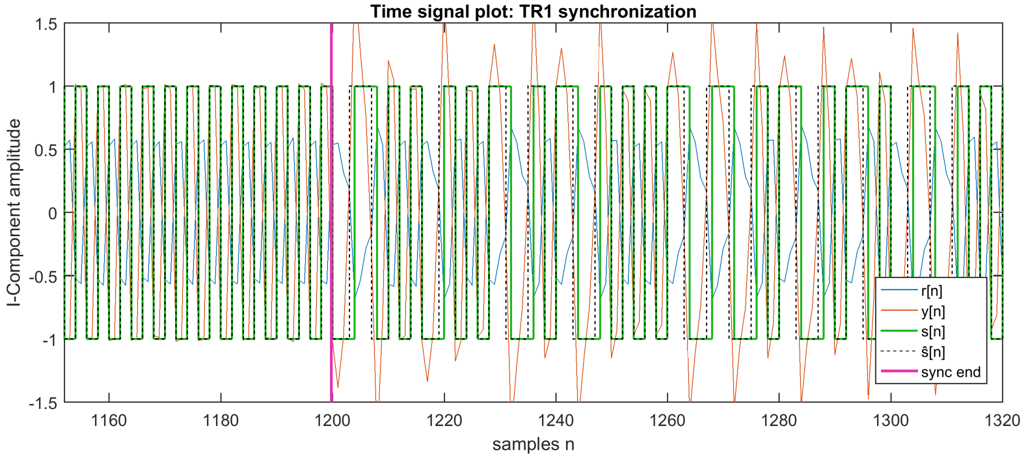


Figure 4.10: Time signals: TR1 synchronization frame followed by random message. $N = 20000$. *Signal errors occur at the start of the message.*

Observation 1 indicates that the equalizer is reacting to changes in the signal. The synchronization signal is restricted to the subcarrier frequency only, it is thus very narrow-band. With the change to regular phase transitions the signal becomes broad-band – it excites a broader frequency range in the spectrum. Since the equalizer is *blind*, it has to react to signals “as they come”. During synchronization it converges to a solution which perfectly accomodates the subcarrier signal. However, at the hard changing point it needs to restart convergence to accomodate the broad-band message signal.

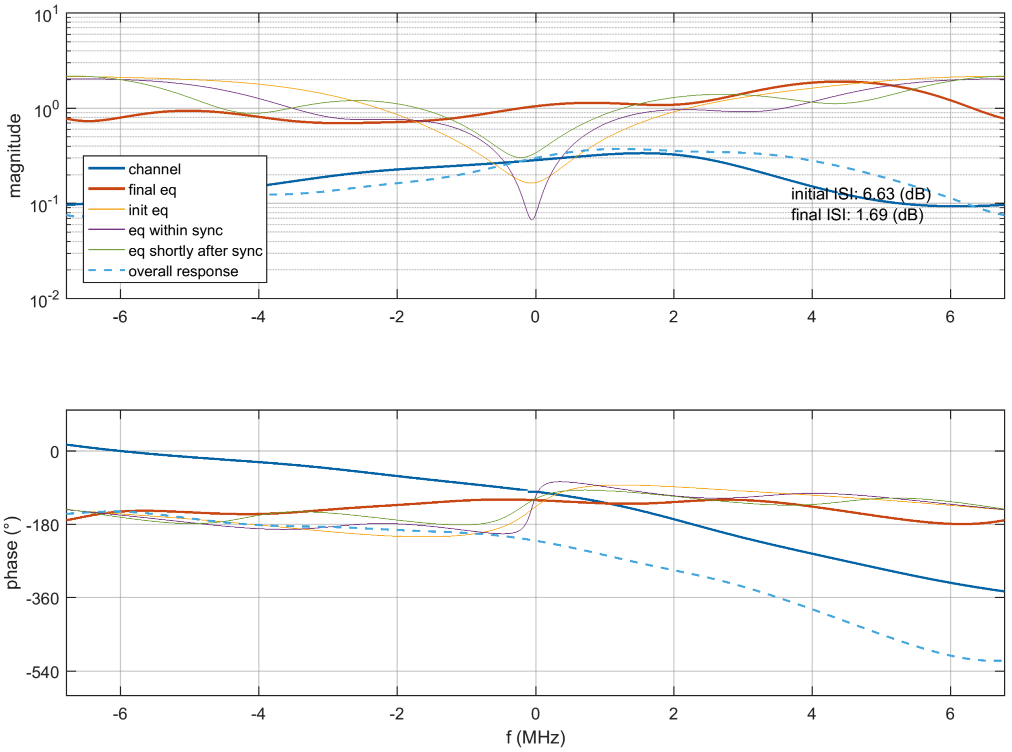
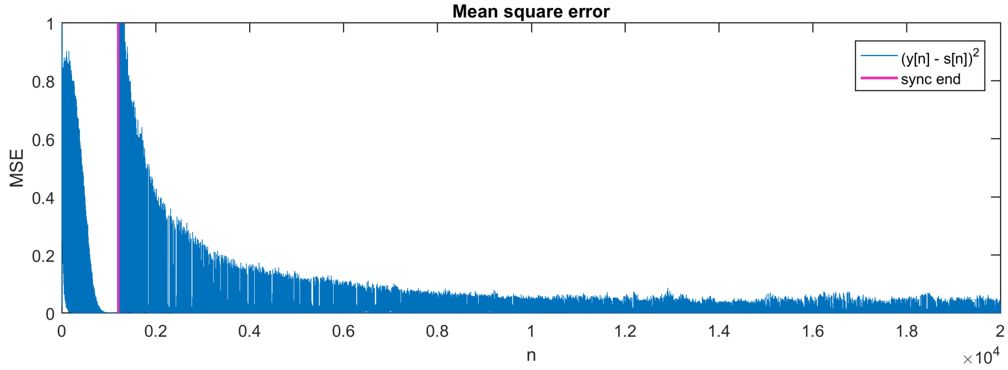
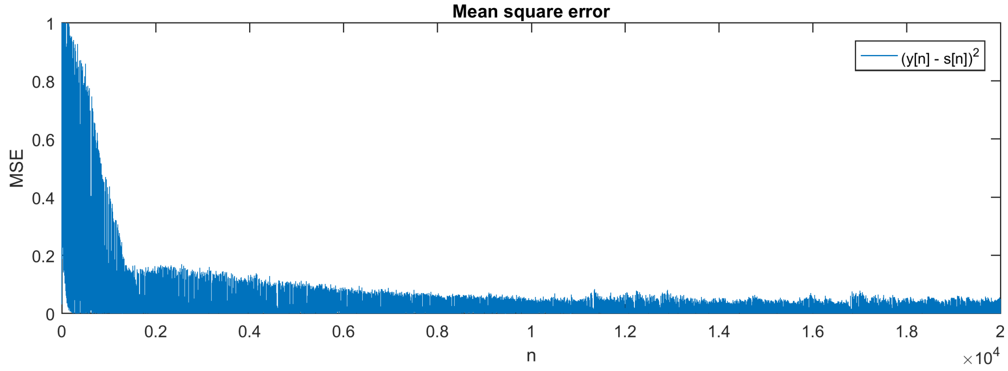


Figure 4.11: Frequency spectra at different stages of TR1 and the message frame. TR1 signal causes the blind equalizer to diverge from its initialization.

The interesting question: Is the blind equalizer able to converge to the intended global minimum, if convergence starts at the coefficient setting “left behind” by the sychronization period instead of the tried-and-true initialization (see Section 4.2.4)? Because the reduced excitation of the TR1 subcarrier only may carry the equalizer away from the global minimum. A visual is given in Figure 4.11. The blue curve is the channel spectrum ($k = 0.43$) with the two resonances already fairly far apart. The yellow line is the initialization of the equalizer. During synchronization, the coefficients cause a steeper spectral notch (violet curve) and a wavy shape left and right of the notch. The green curve represents the equalizer setting shortly after the change from TR1 to message period. It is nowhere near the optimal setting (yet), however, it closely passes the initial setting (yellow line) on its way towards a valid message signal setting. The thick red curve is the final equalizer setting. It roughly equalizes the channels resonances and additionally suppresses some noise around $\pm f_s/2$. The overall system response (dashed cyan curve) does not have a flat magnitude spectrum,[40] which means remaining ISI has been traded for some noise compensation. But more importantly, the phase spectrum shows the correct linear phase shape – which means the equalizer is eventually successful – after a convergence time prolonged by the coefficient deviation due to the TR1 synchronization period.



(a) enabled TR1 synchronization before message



(b) disabled TR1 synchronization (message starts directly)

Figure 4.12: MSE history of wNCMA with and without TR1 synchronization period before the message. $N = 20000$. (a) shows a longer (exponential) decay time than (b) which decays linearly.

The impact on the convergence time is made visible by plotting the mean square error (MSE) over time. Figure 4.12 compares the two cases of (a) enabled TR1 and (b) disabled TR1. In the first case, it is evident that the initial equalizer setting of Section 4.2.4 is not well suited for the incoming narrow-band excitation of the TR1 subcarrier. The equalizer takes several hundred samples to re-adjust. At the changing point (magenta line), the MSE instantly increases and

---

[40] which is responsible for the remaining ISI $> 0\,\mathrm{dB}$

decays exponentially – taking thousands of samples before reaching the noise floor. In contrast, the MSE decreases more rapidly and linearly in the case of a disabled TR1 (see Figure 4.12(b)). This simulation proves experimentally that TR1 causes a deviation in the equalizer settings which lead to a change in convergence behavior and time.

The question posed two paragraphs ago can be answered with a "yes". The equalizer *is* able to converge to the global minimum in spite of the TR1 deviation. Simulations over all k[41] under different noise levels have confirmed this.

A remaining challenge is to reliably detect the "first phase transition" required in the ISO-14443 quotation. This phase transition causes the MSE to jump instantaneously and the equalizer to re-adjust: The most important spot in the entire signal (in terms of successful synchronization and message polarity) is simultaneously the one spot in which the error (MSE and $e[n]$) will be highest.

Last but not least, an idea to resolve this issue shall be given here. Simulations or tests of this idea cannot be conducted due to the time limitations of this thesis' work. The proposed approach is to re-initialize the equalizer shortly before the first phase transition occurs. The few subcarriers remaining before the message then cause only minor deviations in the equalizer settings. The proper initialization stated in Section 4.2.4 is designed for the broad-band excitation case and will thus more likely reconstruct the first phase transition correctly. Looking into the future to anticipate this phase change beforehand sounds "magical", but can be implemented by means of a delay line. Two streams of the received signal $r[n]$ need to be captured in parallel, one of them undelayed, the other one delayed by some samples (known amount). When the undelayed stream reaches the changing point, signal changes occur that are detectable by a (newly designed) change point detector (this detector could resort to change in amplitude measures of $r[n]$ or $y[n]$ for instance). As soon as this detector robustly indicates that the changing point is reached, the equalizer is re-initialized and its input signal switched to the delayed stream. The delayed stream is just about to reach the changing point – and is now well-prepared for the incoming first phase transition.

---

[41] up to $k \approx 0.43$

5

Conclusion

This thesis explores and evaluates the applicability of *blind* adaptive equalization within the specific context of 13.56 MHz RFID systems defined by the ISO-14443 standard. The communication is directed from card to reader (PICC to PCD), and *very high bit rates* (VHBR) of $\geq$ 848 kbit/s are of key interest, as are real-time capabilities. The hypothesis postulates that it is possible to improve the transmission performed in this context by means of compatible blind channel equalization methods.

In order to research the validity of this claim, appropriate steps taken included a thorough system analysis, resulting in a system model for equalizer testing, followed by a selection process of suitable blind equalizers. Modifications are suggested and evaluated alongside the discussion of general viability of blind equalizers. The following section recapitulates and summarizes the most important insights of each chapter, reviews the findings critically and discusses the key results. Finally, an outlook points to remaining open issues and potential future research.

5.1 Summary & Discussion

The first chapter introduces the basics of RFID technology. Inductive (magnetic) coupling between the card's and reader's antennas serves as transmission channel as well as a means of energy transfer. The operating distances are thereby limited to max. 10 cm. The ISO-14443 standard is introduced as a requisite for the hypothesis: possible signal shapes, bitrates, and modulation schemes are stated. Complications that arise from these conditions so far are a) intersymbol interference (ISI) occuring in the given RFID transmission channel – which is b) distance-dependent (coupling) – and last but not least c) an upper limit for increasing data rates (VHBR, comp. Shannon-Hartley theorem, channel capacity). Current equalization methods include matched-filter correlators (sequence detection) and static filters (either analog or ditigal) – which perform well up to 848 kbit/s, but not beyond. Blind equalizers have not been considered in this specific RFID context prior to the research done in this thesis, however, based on the existing knowledge on blind equalization and its successful application in similar contexts, the main expectation here is that the given RFID system will also benefit from such equalization methods.

The second chapter encompasses the analysis of the RFID system. An exemplary circuit schematic is used to explain details of functionality on an electrical level. The internal impedance of the card (PICC) switches between two logic states. These changes are *message signals* and can be "picked up" by the reader (PCD). The distance between the antennas is the one factor with greatest impact on the system's general dynamic behavior. It is equivalently defined as the coupling factor k (2.4), which indicates the strength of mutual magnetic coupling. The coupling effect is intensified via resonance: Both components (card & reader) are tuned to a common resonance frequency (of 13.56 MHz). Coupling strength thus depends on k as well as the shape of the resonance – definable by the *quality factor* Q. The point of *critical coupling* is commonly defined as $kQ = 1$. A study of the RFID system (simulation framework provided by NXP) shows that matching resonances occur at infinite antenna distance. As the antennas close in on each other, their common resonance first becomes maximally wide (point of critical coupling), and then splits up into two separate resonances – one increasing in frequency, the other decreasing (see Figures 2.4 to 2.6). The resulting distortions (frequency-dependent signal gain or attenuation, phase variations around the resonance frequency) are *the* most important defining properties of the RFID system.

Equalization is very costly in terms of implementation (hardware) if done in the *passband*, i.e., using the signal in a form in which it is still modulated onto the carrier frequency (of 13.56 MHz). The equivalent baseband representation is applied instead (with the view of being less costly). Usually, *signals* are translated between these two representations – here, the RFID *system* is subject to this translation. In the baseband (centered around $f = 0\,\text{Hz}$), the translated RFID system becomes *complex-valued* ($\mathbb{C}$) while retaining the general behavior that occurs in the passband. Translation without further model simplifications is successful. A pole-zero analysis of the system reveals that merely three poles and one zero are dynamically changing with respect to k. Reducing the model order based on this insight leads to successful translation in terms of the phase spectrum, however, slight deviations occur in the magnitude spectrum (Figure 2.12).

Since the RFID system model is given as *two systems* actually – one state space model, but *two* switchable states of the card's internal impedance – attempts have been made to combine the two in order to gain a single, compact system model, to which arbitrary message signals could be applied for testing purposes. These attempts ("differential transfer function") have proven not successful and thus have been discontinued. Instead, the focus shifts to the MLS and IRS system identification methods, which are able to yield such a single, compact system model directly (Figure 2.18). This new model (see Section 2.4) is of the well-known ZPK form. It profits from the reduction of the amount of poles and zeros – compared to the former model – but without noticable deviations in this case. A simulation sequence of 100 linearly spaced values of k provides a sufficiently dense grid for a model that covers all dynamic changes in the range of $0.001 < k < 0.55$. Based on the "smoothness" assumption with respect to changing k, IRS measurement errors can be corrected in postprocessing steps (Figure 2.19). Important model properties are summarized in Section 2.4.4. The model validity is discussed in Section 2.4.3: For low k, the model produces accurate signal distortions, identical to those of the original HF simulation (Figure 2.21). With increasing k, weak nonlinear distortions start to occur in the original HF simulation. These nonlinear distortions are not accounted for in the new model. Since nonlinear system modeling rapidly increases in complexity, studies in this direction have been deferred to future research (see Section 5.2). It is, however, assumed that the occuring nonlinear effects are weak enough so that equalizing only the *linear* portion will not lead to a significant loss of validity at higher k. Moreover, the range of k is later recommended to be constrained to approximately $k < 0.43$ anyway (due to insights gained during equalizer evaluation). The overall achievement of Chapter 2 is the detailed knowledge about the system's behavior, as well as a new, simplified system model that is faster, more flexible and designed for the task at hand: studying the performance of various equalizers.

The third chapter examines these equalizers. It starts with the basic concept of a channel-equalizer system cascade (Figure 3.1) and continues with theoretical background on the method of zero-forcing (ZF) equalization, and then the advanced method using the MSE criterion – which leads to the well-established adaptive LMS algorithm. This theoretical information is of vital didactic value for understanding the mechanics of *blind* equalizers. In fact, the algorithm structure of the blind equalizers presented in Chapter 3 is almost identical to the one inherent to LMS. Moreover, LMS is used as a benchmark for evaluation in Chapter 4. Fixed system inverse solutions are pre-computed (offline) via LMS as a reference for achievable optimum equalization. The LMS equalizer – in spite of all its advantages – is itself not an option due to the restrictions in the ISO-14443 standard, which prohibit the use of the *training sequences* required for LMS. This circumstance motivates the use of blind equalizers – which do not require training sequences. Several existing blind algorithms are presented in Chapter 3, which accommodate the additional requirements posed in the hypothesis, i.e., low implementation cost, real-time operability, efficiency, and robustness. Two candidates are singled out: The constant modulus algorithm (CMA) and an algorithm based on Bayesian inference and maximum entropy ("W_{new}"). After a review of blind algorithm behavior (influences of various parameters on their error function surfaces), a modification (MODIFICATION 1) is contributed (Section 3.5), which leads to a new algorithm called wNCMA. This new algorithm is specifically designed to match all given requirements (RFID channel, ISO standard, implementation) in the best conceivable way. It is based on the remaining two candidate algorithms (predominantly W_{new}) and includes the modifications described in Section 3.5. Noteworthy is its capability to suppress errors during message reconstruction that stem from the complex-valued nature of the channel-equalizer cascade.

The fourth chapter describes the simulation setup for the evaluation of the equalizer(s). Details are given about how to initialize the new equalization algorithm properly. Bit error rates (BER) are a widely used means of evaluation in the context of communication systems. Parameter studies have been simulated, relating BER to the normalized signal-to-noise ratio (E_b/N_0), and to the coupling coefficient (k). Results show that acceptable performance of wNCMA in terms of BER is starting in the region of $E_b/N_0 \approx 15$ to 20 dB, depending on the subcarrier frequency (Figure 4.3). The variation over k is quite different for all four subcarriers. Results show that the combined effect of high subcarrier frequency and high $k > 0.4$ leads to increased bit error rates (Figure 4.6). For $k < 0.4$ the wNCMA equalizer performs well, and above $k = 0.5$ it is able to produce a correct but phase-inverted message signal. The region of $k < 0.10$ is more error-prone due to weaker coupling, resulting in a (channel-induced) lower signal gain. This can be seen in all compared equalizers (MMSE, wNCMA and NCMA). An overall performance overview is given in Figure 4.8: The wNCMA outperforms the NCMA in all cases. The wNCMA is very close to the benchmark solution (achievable optimum).

The introduced modification leading to the wNCMA algorithm is also evaluated. Section 4.4 verifies, by a series of simulations, that equalizers with the modification *enabled* clearly favor BPSK output, while their unmodified versions tend to favor arbitrarily complicated – and most importantly false – underlying constellations, which are a sign of ill-convergence and lead to increased bit errors. MODIFICATION 1 thus maximizes signaling robustness by assuming a simple, real-valued BPSK signal as input and output, while allowing the received signal (I/Q) and the equalization to be complex-valued (increased accuracy).

Finally, the influence of the TR1 synchronization period specified in ISO-14443 on the convergence behavior of the equalizer is evaluated. It is shown that the equalizer coefficients diverge slightly due to narrow-band (non-persistent) excitation of the TR1 signal, but are able to re-converge to the correct coefficient setting in all cases of $k < 0.4$ and sufficiently high $E_b/N_0 > 20$ dB. The duration of the transient period is, however, generally longer when TR1 is active. Since – in practice – it will always be active (it is a part of the ISO-14443 protocol), a proposal is made in Section 4.5 on how to deal with this issue.

All in all, this thesis contributes new perspectives and approaches on RFID system modeling, on blind equalizers, and finds a new algorithm for the given application. The hypothesis holds true in the face of the evaluation results. Blind equalization proves successful as an outcome of a prudent fine-tuning process of equalizer parameters. Finally, let it be noted that, when dealing with blind equalizers, the first and foremost objective is precise, contextual adjustment: There is no *universal* blind equalizer — providing the equalizer with auxiliary means in order to make it *less blind* is inherently the task to focus on.

5.2 Outlook

Some of the ideas developed in the course of this thesis could not be pursued (yet). The first one is about the system model: It has been mentioned in Section 2.4.4 that it is a *linear* model, however, evidence shows weak nonlinear effects in the underlying high-frequency model. Nonlinear system models have not been investigated, because the estimated additional effort would dwarf the benefits of having such a model. Nevertheless, research in this direction may be of interest if the nonlinearities occuring in a hardware implementation are more significant than assumed in this thesis. Some keywords in nonlinear modeling are "Volterra Series", "Hammerstein"- and "Wiener"-systems.

As soon as the system (the channel) is assumed to be nonlinear, the equalizer is recommended to be nonlinear, too. In this context, "nonlinear" refers to nonlinear coefficient terms (higher-order kernels) – *not* to the nonlinear methods inherently present in the adaptive algorithms of all blind equalizers. This equalization strategy rapidly increases the implementation complexity (exponential increase in hardware computations and memory). Since this characteristic contradicts the restrictions stated in the hypothesis, such equalizers have not been pursued further.

Another opportunity for possible improvement by nonlinearities has been mentioned in Section 3.3.3. All blind equalizers with a number of coefficients $L < \infty$ suffer from *length-dependent local minima.* Attempts to eliminate such minima involve the extension of blind equalizers to use nonlinearities *with* memory in their cost function (as opposed to memoryless cost functions so far). Paper [40] is the first to deal with this topic, the CRIMNO criterion is introduced. The ideas presented there are a recommended reading: The increasing implementation complexity prevails, but this approach offers possibilities to directly influence and regulate the convergence behavior on the algorithm's cost function surface in return.

One (theoretically considered nonlinear) attempt on equalization *has* been made: The maximum-likelihood sequence estimation (MLSE) approach and especially the deduced, more light-weight *Viterbi algorithm* are a widely used benchmark for other equalizers, because they achieve statistically optimal results and can thus be interpreted as an upper performance bound. A Viterbi equalizer has initially been considered as reference solution within this thesis (not as a valid alternative to blind equalizers, because the channel response needs to be known for this algorithm to work), but has been discontinued: The computational complexity increases exponentially with the length of the channel response.[42] Since the RFID channel response ranges from about 80 taps[43] down to about 10 taps,[44] these constraint lengths are still to high to be practical ([17] stops at a constraint length of 4). The computation time rapidly grows to hours and days for a short, single-pass simulation. Multiple runs for BER Monte-Carlo simulations are beyond question. Nevertheless, a maximum-likelihood detector *after* the proposed blind equalizer may be an attractive alternative to the (basic) slicer implemented in Section 4.1. The recommended reference on MLSE is [23, Ch. 7].

The tracking characteristics (dynamic time-variant RFID system behavior due to changing k) of the proposed blind equalizers have not been evaluated in this thesis. The rate of change is assumed to be sufficiently slow in relation to the sampling frequency (processing speed of the equalizer), so stationarity follows for sufficiently long "blocks" of samples. Paper [13] associates the tracking capability of any SGD-type algorithm with the local curvature of the cost surface. Since the channel's impulse response has been shown to change smoothly with

---

[42] usually called "constraint length"; $2^{(\text{constraint length}-1)}$ states per sample are checked

[43] "significant" samples; for very low k; still truncated — the true response is infinite (IIR)

[44] for very high $k \longrightarrow 0.55$

k (comp. Figure 2.19), the position of the global minimum on the cost surface also changes its position smoothly. SGD-type blind equalizers are thus assessed as likely being able to track time variations in the channel, provided that the equalizer is already close to the global minimum, and provided that the rate of change is sufficiently low. Evidence or simulations are pending.

Recommended topics of investigation during a possible hardware implementation of the proposed scheme are a) double-checking the validity of the RFID system model,[45] and b) should the need arise, re-adjusting the initialization parameters of the equalizer.

---

[45] changes in reader or card circuitry may have significant impact on the blind equalizer's initialization

Bibliography

[1] K. Finkenzeller, *RFID Handbuch*, 7th ed. Hanser Verlag, 2015.

[2] ISO/IEC 14443-1, "Physical characteristics," 2016.

[3] ISO/IEC 14443-3, "Initialization and anticollision," 2016.

[4] ISO/IEC 14443-4, "Transmission protocol," 2016.

[5] ISO/IEC 14443-2, "Radio frequency interface," 2016.

[6] Y. Sato, "A method of self-recovering equalization for multilevel amplitude-modulation systems," *Communications, IEEE Transactions on*, vol. 23, pp. 679 – 682, Jul. 1975.

[7] D. N. Godard, "Self-recovering equalization and carrier tracking in two-dimensional data communication systems," *Communications, IEEE Transactions on*, vol. 28, pp. 1867 – 1875, Dec. 1980.

[8] A. Benveniste, M. Goursat, and G. Ruget, "Robust identification of a nonminimum phase system: Blind adjustment of a linear equalizer in data communications," *Automatic Control, IEEE Transactions on*, vol. 25, pp. 385 – 399, Jul. 1980.

[9] A. Benveniste and M. Goursat, "Blind equalizers," *Communications, IEEE Transactions on*, vol. 32, pp. 871 – 883, Sep. 1984.

[10] G. Picchi and G. Prati, "Blind equalization and carrier recovery using a "stop-and-go" decision-directed algorithm," *Communications, IEEE Transactions on*, vol. 35, pp. 877–887, Jan. 1980.

[11] O. Shalvi and E. Weinstein, "New criteria for blind deconvolution of nonminimum phase systems," *Information Theory, IEEE Transactions on*, no. 36, pp. 312–321, 1990.

[12] Z. Ding and Y. Li, *Blind Channel Equalization and Identification.* New York, NY, USA: Marcel Dekker, Inc., 2001.

[13] C. R. Johnson, P. Schniter, T. J. Endres, J. D. Behm, D. R. Brown, and R. A. Casas, "Blind equalization using the constant modulus criterion: A review," *Proceedings of the IEEE*, vol. 86, pp. 1927–1950, Oct. 1998.

[14] Y. Bai, L. Zhang, and J. Teng, "The application of the CMA blind equalization in UHF RFID system," *2007 IET Conference on Wireless, Mobile and Sensor Networks (CCWMSN07)*, pp. 326–329, Dec. 2007.

[15] R. C. H. van de Beek, M. Ciacci, G. Al-Kadi, P. Kompan, and M. Stark, "A 13.56 mbps PSK receiver for 13.56MHz RFID applications," *2012 IEEE Radio Frequency Integrated Circuits Symposium*, pp. 239–242, Jun. 2012.

[16] V. Berg, J. B. Doré, and F. Frassati, "ISO/IEC 14443 VHBR: Influence of the proximity antennas on the PCD-to-PICC data link performance," *2015 International EURASIP Workshop on RFID Technology (EURFID)*, pp. 81 – 86, Oct. 2015.

[17] J. B. Doré, N. Touati, and F. Pebay-Peyroula, "MLSE detector for beyond VHBR contactless air interface," *2012 IEEE International Conference on RFID-Technologies and Applications (RFID-TA)*, pp. 222–227, Nov. 2012.

[18] M. Sampl, M. Kapper, and E. Leitgeb, “Transfer function of a 13.56MHz RFID channel,” *Loughborough Antennas and Propagation Conference (LAPC)*, pp. 379–382, Nov. 2013.

[19] M. Sampl, “Simulation and evaluation of coding and equalization for 13.56 MHz RFID systems,” Diploma Thesis, Graz University of Technology, May 2007.

[20] G. Lehner, *Elektromagnetische Feldtheorie.* Springer, 2013.

[21] H. Henke, *Elektromagnetische Felder: Theorie und Anwendung.* Springer, 2004.

[22] F. Langford-Smith, *Radio Designer’s Handbook.* Amalgamated Wireless Valve Company PTY. LTD. Australia, 1952. [Online]. Available: http://frank.yueksel.org/other/RCA/Radiotron_Designers-Handbook_Fourth-Edition

[23] J. Barry, E. Lee, and D. Messerschmitt, *Digital Communication*, 3rd ed. Springer, 2004.

[24] M. Meyer, *Signalverarbeitung: Analoge und digitale Signale, Systeme und Filter.* Springer, 2017.

[25] N. Dourdoumas and M. Horn, *Regelungstechnik - Rechnerunterstuetzter Entwurf zeitkontinuierlicher und zeitdiskreter Regelkreise.* Pearson, 2004.

[26] S. Haykin, *Adaptive Filter Theory*, 3rd ed. Prentice Hall, 1996.

[27] S. Golomb, *Shift register sequences.* Holden-Day, 1967.

[28] G.-B. Stan, J.-J. Embrechts, and D. Archambeau, “Comparison of different impulse response measurement techniques,” *J. Audio Eng. Soc*, vol. 50, no. 4, pp. 249–262, 2002.

[29] M. T. G. Hughes and A. R. M. Noton, “The measurement of control system characteristics by means of a cross-correlator,” *Proceedings of the IEE - Part B: Electronic and Communication Engineering*, vol. 109, no. 43, pp. 77–83, Jan. 1962.

[30] J. Borish and J. B. Angell, “An efficient algorithm for measuring the impulse response using pseudorandom noise,” *J. Audio Eng. Soc*, vol. 31, no. 7/8, pp. 478–488, 1983.

[31] D. D. Rife and J. Vanderkooy, “Transfer-Function Measurement with Maximum-Length Sequences,” *J. Audio Eng. Soc*, vol. 37, no. 6, pp. 419–444, 1989.

[32] J. Vanderkooy, “Aspects of mls measuring systems,” *J. Audio Eng. Soc*, vol. 42, no. 4, pp. 219–231, 1994.

[33] C. Dunn and M. J. Hawksford, “Distortion Immunity of MLS-Derived Impulse Response Measurements,” *J. Audio Eng. Soc*, vol. 41, no. 5, pp. 314–335, 1993.

[34] M. Wright and J. Vanderkooy, “Comments on ’Aspects of MLS Measuring Systems’,” *J. Audio Eng. Soc*, vol. 43, no. 1/2, pp. 48, 49, 1995.

[35] N. Ream, “Testing a 2-input linear system with periodic binary sequences,” *Electrical Engineers, Proceedings of the Institution of*, vol. 114, no. 2, pp. 305–307, Feb. 1967.

[36] J. O. Smith, *Introduction to Digital Filters with Audio Applications.* [online] https://ccrma.stanford.edu/~jos/filters/: W3K Publishing, CCRMA Stanford University, 2007.

[37] M. Pinchas and B. Z. Bobrovsky, “A maximum entropy approach for blind deconvolution,” *Signal Processing*, vol. 86, pp. 2913–2931, Oct. 2006.

[38] A. V. Oppenheim, R. W. Schafer, and J. R. Buck, *Discrete-time Signal Processing (2nd Ed.).* Prentice-Hall, 1999.

[39] S. Haykin, *Blind Deconvolution.* Prentice Hall, 1994.

[40] Y. Chen, C. Nikias, and J. Proakis, “CRIMNO: criterion with memory nonlinearity for blind equalization,” *Systems and Computers Asilomar Conference on Signals*, pp. 694 – 698, Dec. 1991.

[41] C. Papadias, “Methods for blind equalization and identification of linear channels,” PhD thesis, Ecole Nationale Superieure des Telecommunications (IMT, Paris), Mar. 1995.

[42] S. Kay, *Fundamentals of Statistical Signal Processing: Detection theory*, ser. Prentice Hall Signal Processing Series. Prentice-Hall, 1998.

List of Figures

List of Tables

Acronyms & Abbreviations

a_j	Denominator coefficients of a polynomial system model
a_{ZF}	Scaling gain factor that appears in ZF equalization
$\boldsymbol{A}$	System matrix of a general-form state space model
ACF	Auto-Correlation Function
AGC	Automatic Gain Control (circuit regulating the input gain at the receiver)
AR(X)	Autoregressive process or model of order (X)
AWGN	Additive White Gaussian Noise (channel disturbance model)
b_j	Numerator coefficients of a polynomial system model
b_l	Binary sequence (see also l)
$b(t)$	Continuous-time message signal (associated with binary sequences)
$\boldsymbol{B}$	Input matrix (or vector) of a general-form state space model
$\mathbf{B}$	Magnetic field (vector field)
BER	Bit Error Rate
BGR	Algorithm familiy named after their inventors (Benveniste, Goursat, Ruget)
BPSK	Binary Phase Shift Keying
$c[n]$	Equalizer coefficients (also in vector notation: $\boldsymbol{c}$, $\boldsymbol{c}_{opt}$, $\boldsymbol{c}[n]$)
$C(z)$	Z-domain representation of the equalizer
C	Crest factor
C_i	Capacitance (network theory)
$\boldsymbol{C}$	Output matrix (or vector $\boldsymbol{c}^T$) of a general-form state space model
$\mathbf{C}$	Highlighted constant within the Bussgang theorem
CMA	Constant Modulus Algorithm
CRIMNO	CRIterion with Memory NOnlinearity (proposition for blind equalizers)
$d[n]$	Discrete-time training (reference) signal (in MMSE equalizers)
$\boldsymbol{D}$	Feed-through vector (or scalar d) of a general-form state space model
dB	Decibel (logarithmic measurement unit)
DSP	Digital Signal Processor (hardware) or Processing (method)
$e[n]$	Discrete-time algorithm error signal (in adaptive equalizers)
E_b/N_0	Energy of one bit normalized by noise spectral density ("SNR per bit")
$\mathbf{E}$	Electric field (vector field)
EM	Electro Magnetic ...
EQ	Equalizer
f_c	Carrier Frequency
f_{sub}	Subcarrier Frequency
f_s	Sampling Frequency (of DSP unit)
FFT	Fast Fourier Transform
FIR	Finite Impulse Response (finite in length, comp. IIR)
FPGA	Field-Programmable Gate Array (hardware)
$\mathbf{g}$	System gain (ZPK model)
$g(\cdot)$	1. Nonlinear function substitute (context: blind equalizers, Bussgang theorem) 2. Function used to demonstrate symmetry properties of the Fourier transform
$G(s)$	Laplace domain representation of a system G
$h(t)$	Continuous-time impulse response of a system h
$h[n]$	Discrete-time impulse response of a system h
$H(f)$	Fourier spectrum of a system H
$H(z)$	Z-domain representation of a system H

HF	High-Frequency (assoc. with passband, here: region around 13.56 MHz)
HOS	Higher-Order Statistics (e.g., moments, cumulants, ...)
Hz	Hertz (derived SI frequency unit)
$i(t)$	Continuous-time electric current
$i_{1;2}[m]$	Intermediate signal used for BER computation; discrete-time; f_{sub} rate $[m]$
I	1. Current (network theory) 2. In-phase component (communication signalling context)
IC	Integrated Circuit
IEC	International Electrotechnical Commission
i.i.d.	Independent and identically distributed (property of random variables)
IIR	Infinite Impulse Response (infinite in length, comp. FIR)
IRS	Inverse Repeat Sequence
ISI	Inter-Symbol Interference
ISO	International Standardization Organization
j	1. Imaginary unit ($j^2 = -1$) 2. also used as index ($\sum$-formulas)
k	Coupling factor
K	1. Kurtosis (4^{th} standardized moment; see Section 3.3.1) 2. Gain parameter within the continuous ZPK model
l	Position or rate index
L	Represents an integer number indicating a length
L_i	Inductance (network theory)
LMS	Least Mean Squares (algorithm)
LTI	Linear, Time-Invariant (system properties)
M	1. Mutual inductance 2. Number of Monte-Carlo repetitions (during BER simulation)
M-PAM	see PAM, "M" specifies the alphabet order
MA(X)	Moving-Average process or model of order (X)
MAP	Maximum a posteriori (probability estimation method, Bayesian statistics)
MATLAB	Numerical computing programming language (software)
MHz	Megahertz (derived SI frequency unit, mega = 10^6)
ML	Maximum Likelihood (probability estimation method, Bayesian statistics)
MLSE	Maximum Likelihood Sequence Estimation / Estimator
MLS	Maximum Length Sequence (not to be confused with MLSE)
MMSE	Minimum Mean-Square Error (minimized MSE, also name of minim. technique)
MSE	Mean-Squared Error (error definition, criterion)
N	Represents an integer number indicating a length
NRZ-L	Non-Return-to-Zero Level
NCMA	Normalized CMA (see Modification 2)
NLMS	Normalized LMS (see Modification 2)
$o[n]$	Discrete-time impulse response assoc. with the *overall* system (see (3.26))
p	Exponent index in Section 3.3.1
$p_i/\mathbf{p}_i$	System pole (ZPK model)
$\boldsymbol{p}$	Crosscorrelation vector (see MMSE equalizers)
$P(f)$	Hilbert filter (in the Fourier frequency domain)
PAM	Pulse Amplitude Modulation (digital modulation method)
PAPR	Peak-to-Average-Power Ratio
PCD	Proximity Coupling Device (Reader)
PDF	Probability Density Function (continuous random variables)

PMF	Probability Mass Function (discrete random variables)
PICC	Proximity Integrated Circuit Card
PRBS	Pseudo-Random Binary Sequence
PSD	Power Spectral Density (also as $P_{xx}(f)$ = PSD of x)
PSK	Phase Shift Keying (digital modulation method)
Q	1. Quality (attribute of resonant networks) 2. Quadrature-phase component (communications signaling context) 3. Function representing a decision device (slicer) operation: $Q(\cdot)$ (Section 3.3.1)
Q	Integer up-/downsampling factor (see Figure 4.1 and Section 4.1)
QAM	Quadrature Amplitude Modulation (digital modulation method)
QPSK	Quadrature Phase Shift Keying (digital modulation method)
$r_{xx}(\cdot)$	Autocorrelation function/sequence of the signal x
$r[n]$	Discrete-time channel output ("received") signal (including additive noise)
R_{mod}	Resistor switchable between two states for modulation purposes
$\boldsymbol{R}$	Autocorrelation matrix (associated with $r[n]$, resp. $\boldsymbol{r}[n]$)
RFID	Radio Frequency Identification
RLS	Recursive Least-Squares; variant of LMS (adaptive algorithm)
RMS	Root-Mean-Square (derived approximate measure of signal power)
s	Laplace variable (complex-valued)
$s(t)$	Continuous-time signal (general)
$\tilde{s}(t)$	Continuous-time, complex-valued signal envelope (up-/downconversion)
$s[n]$	Discrete-time channel input signal
$\hat{s}[n]$	Estimated/reconstructed version of $s[n]$
$\mathbf{S}$	Reference surface (to calculate the magnetic flux Φ)
$\mathbf{S}^*(-\mathbf{f})$	Highlighted spectral mirror copy (context: up-/downconversion)
SGD	Stochastic Gradient Descent (optimization method)
SISO	Single-Input-Single-Output (system property)
SNR	Signal-to-Noise Ratio
T	Signal period
TF	Transfer Function
TR0	Guard time interval (defined in ISO-14443)
TR1	Synchronization time interval (defined in ISO-14443)
UHF	Ultra High Frequency (decimeter band: 300 MHz – 3 GHz)
$v(t)$	Continuous-time electric voltage
V	Voltage (network theory)
VHBR	Very High Bit Rate(s)
VHDR	Very High Data Rate(s)
$w[n]$	Discrete-time AWGN signal
$\tilde{w}[n]$	Discrete-time *filtered* AWGN signal (post-EQ shaped noise)
W_{new}	Algorithm name originating from [37]; transl. nomenclature: $W_{new}[n] \longrightarrow e[n]$
wNCMA	well-behaved (W_{new}-/CMA-related) normalized algorithm (thesis proposition)
$x[n]$	Discrete-time channel output signal (without noise added yet)
$y[n]$	Discrete-time equalizer output signal
$z_i/\mathbf{z}_i$	System zero (ZPK model)
ZF	Zero-Forcing (equalization approach)
ZOH	Zero-Order-Hold (upsampling strategy)
ZPK	Zero-Pole-Gain (model)

α	1. Shift variable (Laplace transform frequency shift property)
	2. Small, non-negative, constant offset (see MODIFICATION 2 and Section 4.2.3)
β	Phase rotation factor (equalizer initialization)
δ	1. Dirac delta (continuous) or 2. Kronecker delta (discrete)
Δ	1. Integer delay (context: ZF equalization)
	2. Difference operator (as in $\Delta\boldsymbol{c}$)
μ	Stepsize (parameter of SGD-type algorithms)
$\xi[n]$	Misalignent of equalizer coefficients (math. ISI representation during convergence)
Π	1. (as $\Pi(\cdot)$) distribution function (PDF/PMF)
	2. [Product formula] (ZPK model)
σ	1. (as σ^2) variance of random variable or process (e.g., Gaussian)
	2. (as in $s = \sigma + j\omega$) real part of Laplace variable s
τ	Lag (shift) parameter in the correlation function
Φ	Magnetic flux
Ψ	Cost function (of SGD-type algorithms)
Ψ'	Error function (of SGD-type algorithms); gradient/derivative of Ψ
ω	Angular frequency [rad/s]
Ω	Ohm (physical unit of resistance)
$\mathbb{C}$	Symbol for the set of complex numbers
$\mathbb{N}$	Symbol for the set of natural numbers
$\mathbb{R}$	Symbol for the set of real numbers
$*$	Convolution operator symbol (asterisk)
∇	Nabla operator $\Rightarrow$ *grad* (gradient)
$\nabla_{\boldsymbol{c}}$	Gradient with respect to $\boldsymbol{c}$
$\nabla\times$	Nabla operator & cross product $\Rightarrow$ *curl* (see Maxwell equations)
$\mathrm{E}\{\cdot\}$	Expectation operator
$\mathcal{F}\{x(t)\}$	Fourier transform operator
$\mathcal{F}^{-1}\{H(j\omega)\}$	inverse Fourier transform operator
$\mathcal{L}\{s(t)\}$	Laplace transform operator
$\mathcal{L}^{-1}\{G(s)\}$	inverse Laplace transform operator
$sgn(\cdot)$	Signum function
$\mathrm{Re}\{\cdot\}$	Real part of a complex variable
$\mathrm{Im}\{\cdot\}$	Imaginary part of a complex variable
$(\cdot)^*$	Complex conjugation
$(\cdot)^T$	Vector/Matrix transpose
$(\cdot)^H$	Hermitian vector/matrix transpose
$(\cdot)^{-1}$	Matrix inversion (or system inverse)
$\lvert\cdot\rvert$	Absolute value operator
$\hat{}$	Estimated entity
$\lll$	very small (number / comparison)
$\ggg$	very large (number / comparison)

Appendix A

```
close all; clear; clc;
%% Downconversion of RLC model
%  using symbolic variables, transfer functions and zero/pole selection
%  Choose: A) Shift NON-reduced  (reduce == 0)
%          B) Shift reduced (1-pole simplified baseband model)
reduce = 1;

%% Transfer Function for RLC series resonant circuit, for output u_R
R = 10;
L = 1e-3;  % 3 mH
C = 1e-7;  % 100 nF

%  Free resonance
w0 = 1/sqrt(L*C);

%  Damping influence on resonance
delta = R/(2*L);
wE = sqrt(w0^2 - delta^2);

%  Definition: s is a "tf" object used to declare transfer functions in s
s = tf('s');
Gs = R/L * s/(s^2 + R/L * s + 1/(L*C));

figure; bodeplot(Gs); grid on;
title('Bode Plot: Serial RLC Circuit, u_R');


%% Conversion to symbolic variables
clear s; syms s

%  Re-definition: s is now a "sym" variable object
G = R/L * s/(s^2 + R/L * s + 1/(L*C));


%% Shift to equivalent baseband
%  Substitute s = s - jw0
G_new = simplify(subs(G, s, s + 1i * w0));

%  Evaluate numerator and denominator after the shift
[num, den] = numden(G_new);

SYMpoles = solve(den);          % analytic solution (exact)
poles = double(solve(den));     % solution converted to double

SYMzeros = solve(num);
zeross = double(solve(num));


%% Select the poles and zeros that are to be kept
%  Poles: The RLC passband is symmetric, thus has 2 conjugated poles.
%         Keep only the one pole that was shifted to DC.
relevant_poles = sort(poles(imag(poles) ~= -w0));
keep = min(relevant_poles);
remove = max(relevant_poles);


%% Re-Write as transfer function again
if reduce == 0
    % NONREDUCED:  directly uses all evaluated poles and zeros
    gain = R*1e3;
    G_baseband = tf(zpk(zeross,transpose(poles),gain));   % WORKING!
else
    % REDUCED:  uses only the remaining pole; compensates gain
    G_baseband = tf([0 (R*1e-2)*zeross/2], [1 -keep]);
    % Shift the phase -90 degrees (see Hilbert transform)
    G_baseband = G_baseband * exp(1i*pi/2);
end

G_baseband

%% Several plots
figure; pzmap(G_baseband); xlim([1.2*real(poles(1)) 1000]);
title('Pole-Zero Map (Laplace plain) of system after the shift');

% - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
wvec = -2*w0:1e2:2*w0;

[mag, phase] = bode(G_baseband, wvec);
mag = mag(:);
phase = phase(:);

[mag1, phase1] = bode(Gs, wvec);
mag1 = mag1(:);
phase1 = phase1(:);


f1 = figure;
h1 = subplot(211);
semilogy(wvec, mag1); hold on
semilogy(wvec, mag);
plot(gca,[-w0 -w0],[1e-3 1e3],'--k');
plot(gca,[w0 w0],[1e-3 1e3],'--k');
plot(gca,[0 0],[1e-3 1e3],'--','color',[0.6 0.6 0.6],'linewidth',0.8);
```

```
hold off
grid on
str1 = ’Frequency Response of G(s) $\rightarrow$ G(s-j$\omega_0$)’;
str2 = ’(equiv. baseband $\in$ C)’;
title(h1,str1,’interpreter’,’latex’,’fontsize’,14)
ylim([0.05/R*max([max(mag) max(mag1)]) 1.1*max([max(mag) max(mag1)])])
ylabel(’Magnitude (dB)’)
legend(’|G(s)|’,’|G(s-j\omega_0)|’,’location’,’southeast’);
h2 = subplot(212);
plot(wvec, phase1); hold on
plot(wvec, phase);
plot(gca,[-w0 -w0],[-360 500],’--k’);
plot(gca,[w0 w0],[-360 500],’--k’);
plot(gca,[0 0],[-360 500],’--’,’color’,[0.6 0.6 0.6],’linewidth’,0.8);
hold off
title(h2,str2,’interpreter’,’latex’,’fontsize’,14)
grid on
ylim([-220 220]);
yticks([-180 0 180 360]);
ylabel(’Phase (deg)’)
xlabel(’Frequency (rad/s)’)
legend(’arg(G(s))’,’arg(G(s-j\omega_0))’,’location’,’southeast’);
all_ha = findobj( f1, ’type’, ’axes’, ’tag’, ’’ );
linkaxes( all_ha, ’x’ );
set(gcf,’PaperPosition’,[0.25 0.25 18 12]);

% - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

%% Discretize: G(s) to G(z)
fs = w0/2/pi;
Ts = 1/fs;
G_z = c2d(G_baseband, Ts, ’matched’);   % ’tustin’

G_z


[B,A] = tfdata(G_z);
B = B{:};
A = A{:};

figure;
zplane(B, A);
xlim([-1 1])
ylim([-1 1])
titlstr = strcat(’Pole-Zero Map (Z-plane) of the baseband region\n’, ...
                 ’sampled at fs = %.2f [Hz] / Ts = %.4f [ms]’);
title(sprintf(titlstr,fs,Ts*1000));


%% Impulse Response
imp = [1; zeros(40,1)];

IR = filter(B,A,imp);

figure;
plot(real(IR)); hold on;
plot(imag(IR)); hold off;
if reduce == 1
    title(’Complex Impulse Response of simplified Metrics Model’);
else
    title(’Complex Impulse Response of non-reduced Metrics Model’);
end
legend(’Re’,’Im’)
grid on;
xlim([0 numel(imp)]);
```

MATLAB script for downconversion of RLC series resonance circuit

Appendix B

```
close all; clear; clc;
%% Complex transfer function estimation example
%  generic complex MATLAB data sample (unsymmetric frequency response)
load cpxbp.mat

%  turn to complex impulse response (for prony & stmcb)
h = ifft(H1,’nonsymmetric’);

%  polynomial orders
na = 4;
nb = 4;

%  methods
[B1,A1] = prony(h,nb,na);
[B2,A2] = stmcb(h,nb,na,20);   % 20 iterations
[B3,A3] = invfreqz(fftshift(H1),F1*pi,’complex’,nb,na);

[z1,p1,g1] = tf2zpk(B1,A1);
[z2,p2,g2] = tf2zpk(B2,A2);
[z3,p3,g3] = tf2zpk(B3,A3);

% compare coefficients or poles and zeros:
B_compare = [B1; B2; B3];
A_compare = [A1; A2; A3];
z_compare = transpose([z1 z2 z3]);
p_compare = transpose([p1 p2 p3]);
g_compare = transpose([g1 g2 g3]);

figure;
s1 = subplot(211);
plot(F1+1,abs(H1)); hold on; grid on;
ylabel(’Magnitude’);
s2 = subplot(212);
plot(F1+1,unwrap(angle(H1))/pi*180); hold on; grid on;
xlabel(’normalized frequency (full circle)’);
ylabel(’Phase’);
[H1,W1] = freqz(B1,A1,1024,’whole’,2);
[H2,W2] = freqz(B2,A2,1024,’whole’,2);
[H3,W3] = freqz(B3,A3,1024,’whole’);
subplot(s1);
plot(W1,abs(H1));
plot(W2,abs(H2),’g--’);
plot(W3/pi,abs(H3),’k:’);
legend(’H_{Original}’,’H_{Prony}’,’H_{Steiglitz}’,’H_{invfreqz}’,’interpreter’,’latex’);
subplot(s2);
plot(W1,unwrap(angle(H1))/pi*180);
plot(W2,unwrap(angle(H2))/pi*180,’g--’);
plot(W3/pi,unwrap(angle(H3))/pi*180,’k:’);
linkaxes([s1 s2], ’x’);

figure;
[HZ,HP,Hl] = zplane([],[]); hold on;
hasbehavior(HZ,’legend’,false);
hasbehavior(HP,’legend’,false);
hasbehavior(Hl,’legend’,false);
scatter(real(p1),imag(p1),’xr’);
scatter(real(p2),imag(p2),’xg’);
scatter(real(p3),imag(p3),’xk’);
scatter(real(z3),imag(z3),’ok’);
scatter(real(z2),imag(z2),’og’);
scatter(real(z1),imag(z1),’or’);
legend(’Prony’,’Steiglitz’,’invfreqz’);
title(sprintf(’Pole (x) Zero (o) Map’));
```

MATLAB code comparing prony, stmcb and invfreqz in complex-coefficient transfer function estimation