Master Thesis

Czech
Technical
University
in Prague

F3

Faculty of Electrical Engineering
Department of Radio Engineering

Channel Estimation and Network Coded Modulation for Parametric H-MAC Channel in WPNC Radio Networks

Petr Hron

Supervisor: prof. Ing. Jan Sýkora, CSc.
Field of study: Signal Processing
May 2019

Acknowledgements

Declaration

I declare that I completed the presented thesis independently and that all used sources are quoted in accordance with the Methodological instructions that cover the ethical principles for writing an academic thesis.

In Prague, 24. May 2019

Prohlašuji, že jsem předloženou práci vypracoval samostatně, a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolkých závěrečných prací

V Praze, 24. May 2019

....

Abstract

This thesis deals with the channel state estimation (CSE) in Wireless Physical Layer Network Coding (WPNC) radio networks. A non-pilot based estimator for a two-source H-MAC channel is presented. The channel phase parametrization is estimated based on hierarchical data decisions.

We present the derivation of an maximum likelihood (ML) hierarchical data decision aided estimator and analyze its theoretical performance limit and properties of the resulting estimation metric.

Based on the theoretical results we propose specific estimation algorithms for scenarios with both constant and varying channel phase parametrization and evaluate their performance using a simulation. Finally, using the CSE together with a hierarchical demodulator and LDPC decoder, we compose a complete working receiver front-end and evaluate its performance in terms of bit error rate.

Keywords: WPNC, iterative channel estimator, hierarchical MAC channel

Supervisor: prof. Ing. Jan Sýkora, CSc.

Abstrakt

Tato práce se zabývá estimací stavu kanálu v radiových sítích s WPNC. Cílem je estimátor fáze hierarchického MAC kanálu se dvěma zdroji, který nevyžaduje použití pilotních signálů. Estimace je prováděna pouze na základě znalosti hierarchických symbolů.

Práce obsahuje odvození ML hierarchického estimátoru a jeho analýzu ve smyslu stanovení teoretického limitu rozptylu odhadů a vlastnosti estimační metriky.

Na základě teoretických výsledků jsou navrženy konkrétní estimační algoritmy pro případy konstantní i proměnné fáze kanálu. Jejich vlastnosti jsou analyzovány pomocí numerických simulací. Na závěr je pomocí navrženého estimátoru, hierarchického demodulátoru a LDPC dekodéru utvořen kompletní frontend přijímače a na základě simulace vyhodnocená jeho bitová chybovost.

Klíčová slova: WPNC, iterativní estimátor kanálu, hierarchichý MAC kanál

Překlad názvu: Estimace kanálu a síťově kódované modulace pro parametrický H-MAC kanál v WPNC rádiových sítích

Contents

Figures

Tables

Chapter 1

Introduction

Wireless Physical Layer Network Coding is a concept addressing the physical layer of dense radio networks. In such networks with many nodes, who all wish to communicate at the same time, the classical concept of orthogonal separation has little application. WPNC on the other hand tries to make use of the crowded spectrum and deliberately allows multiple sources to transmit in the same channel. The interacting signals are assumed to get relayed multiple times before arriving at the destination. From the superposition of multiple signals, the relay is typically not able to decode the original messages. It applies a so-called hierarchical decoding, by which it decodes only some many-to one function of the original messages and this is what gets passed on to further relaying nodes. At the destination however, the original message must reliably decodable. For this to work, it is crucial that the individual nodes are aware of the whole network topology and can choose such signal processing strategies that ensure the overall end-to-end solvability. This way one message can travel trough multiple paths and combine at its destination. This technique increases the overall network throughput and can use the radio resources more efficiently.

For the hierarchical decoding at the relay, the knowledge of the (many to one) channel state is important. Mainly because the combined constellation as perceived by the relay, depends on the relative channel parametrization in a nonlinear way. This work focuses on the channel state estimation in such a scenario, where multiple transmitting nodes share one radio resource. For a time varying channel, the main difficulty remains in the signal superposition, where it is not possible to separate individual sources and thus pilot based approaches cannot be applied.

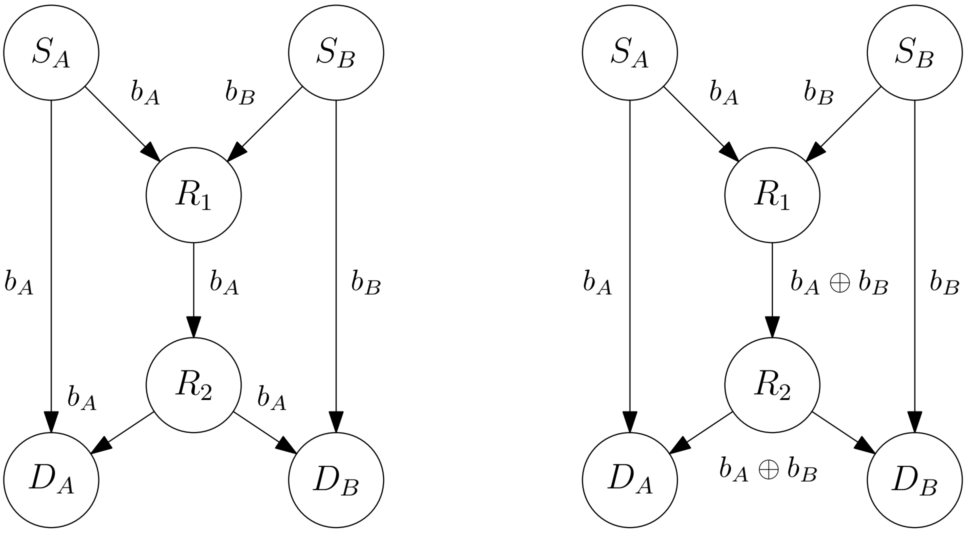
Chapter 2

Fundamental Principles of WPNC

Wireless Physical Layer Network Coding (WPNC) is a new paradigm of network communication. It has the potential to be superior to the classical approach in terms of throughput and efficiency. WPNC combines two major ideas namely *network coding* (NC) and its application directly on the wireless physical layer. This chapter is based on [3], where you can find additional details and further aspects of WPNC, not covered in this brief introduction.

2.0.1 Network Coding

Let us suppose we have a classical wired network with data source nodes, relay nodes and data destination nodes, all connected by metallic links. Such a network can be represented in terms of a graph. Individual links are represented by edges, weighted by the respective link capacities. In networks with one source node and one destination node, the maximal throughput can be found as the capacity of the minimal cut, separating the source node from the destination node. This is known as the *Max-flow min-cut* theorem. In this case the relay nodes play a role of simple switches with a direct link from inputs to outputs. If, however, there are multiple destination nodes which receive the same data, the maximum flow cannot be achieved using such simple relay nodes. For such cases, we need relays which compute and pass on some function of their input data. This principle can be nicely demonstrated on a so-called *butterfly* network, which comprises of two source nodes S_A, S_B, two relays R_1, R_2 and two destination nodes D_A, D_B. Their interconnection is shown in Figure 2.1. All links are of equal capacity, allowing them to pass



(a) : Conventional case.

(b) : Whit network coding.

Figure 2.1: Butterfly network

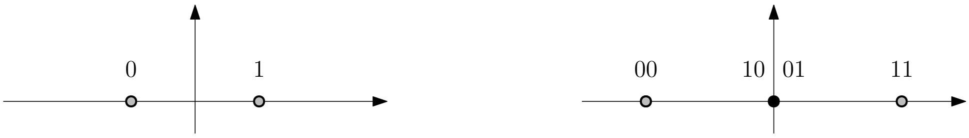
one message. Let us suppose that each source node has a binary message of length m. We denote them $b_A, b_B \in \mathcal{F}_2^m$ respectively. Both messages need to be transferred to each destination D_A, D_B. In a network without NC, the capacity of the link (R_1, R_2) would force R_1 to choose only one of its incoming messages to be forwarded. R_2 would then duplicate its incoming message to both outgoing links. From Figure 2.1a we clearly see, that by using this approach, we consume the whole link capacity and are not able to transfer b_B to D_A. If we use NC instead, R_1 forwards a function of both its incoming messages. In this particular case we use a simple sum on $\mathcal{F}_2^m$ which is the same as bit-wise XOR. Using the fact that $x \oplus x = 0$, the message b_B can be obtained from the inputs of D_A as their sum since $b_A \oplus (b_A \oplus b_B) = (b_A \oplus b_A) \oplus b_B = b_B$. The same holds for D_B as well. On this simple example we can demonstrate how we can increase the throughput of a complex network with multiple sources and destinations by the involvement of NC.

2.0.2 Wireless Network

In a wired network we have precisely defined point to point links, formed by individual wired connections. In a wireless network however, the electromagnetic waves from the transmitters propagate in all directions and combine together at the receiver's antenna. In the classical approach this is perceived as a negative phenomenon and the goal is to separate the useful part of the signal from the rest, the interference. This is accomplished by a division of the signal space into orthogonal subspaces in time, frequency or by means of spatial separation. The orthogonal subspaces are then assigned to individual point to point links. This is in agreement with the frequently used layered model paradigm. According to this model, the communication is divided into multiple sub-tasks, which are then addressed independently on different levels of abstraction. For example in the OSI model, we have the physical layer, which is responsible for the point to point transmission over the physical medium, and the network layer which, is in charge of routing and traffic control. This strict isolation of different parts of the problem leads, in a general case, to an inefficient solution. The relay R_1 from Section 2.0.1 would, according to the layered model, first need to get b_A, b_B separately at the physical layer (using two orthogonal signal subspaces) and then separately apply the XOR function at the routing layer. In the case of R_2, also two orthogonal subspaces would be needed to form the links (R_2, D_A) and (R_2, D_B). WPNC, on the other hand, addresses the problem of network routing directly on the physical layer. This means that the knowledge of the network topology is available at the physical layer and thus the radio resources can be used more efficiently.

This principle can be demonstrated on the operation of R_1 in the butterfly network. Let us suppose that an uncoded BPSK modulation is used on both sources S_A, S_B. Further we assume a perfect symbol timing synchronization between them and ideal radio channels, which do not transform passing signals. If we allow both sources to transfer one bit in the same radio resource, both signals interfere at the receiving antenna of R_1. In Figure 2.2a we see a classical BPSK constellation as transmitted from the sources. On the relay however, due to signal superposition, we get a constellation as shown in Figure 2.2b. Clearly it is not possible to recover the two-bit combination which was originally sent. If the relay receives the constellation point 0, it cannot decide whether the originally sent combination was $(1, 0)$ or $(0, 1)$. However the physical layer is aware of the network topology and the used NC, so it does not attempt to decode both individual messages but only their NC function (in our case the XOR), which can be done with certainty. As shown in Table 2.1, in our case it is enough to decide whether we received a 0 or a ± 2. This way we do not need two orthogonal subspaces to accomplish a transmission from multiple sources to one relay.

In the case of R_2 WPNC would also bring a benefit. Because the physical layer is aware of the network topology it can exploit the fact that the message for both destination nodes is the same and it is sufficient to send it only once, using one orthogonal subspace. In the classical approach, the physical layer is not aware of the surroundings and would use one orthogonal resource for each link $(R_2, D_1), (R_2, D_2)$.



(a) : Single source.

(b) : Two superposed sources.

Figure 2.2: BPSK constellation

b_A	b_B	$b = b_A \oplus b_B$	$s(b_A)$	$s(b_B)$	$s(b) = s(b_A) + s(b_B)$
0	0	0	-1	-1	-2
0	1	1	-1	1	0
1	0	1	1	-1	0
1	1	0	1	1	2

Table 2.1: Two source data combinations and corresponding BPSK constellation points. b_A, b_B denote the source bits, b the target bit and $s(.)$ a BPSK constellation mapper.

2.1 Definitions and Network Model

Now we have an intuitive understanding of the basic ideas behind WPNC. In this section we provide the necessary definitions and the overall network model to develop a good understanding of the whole system.

2.1.1 Network Model

The network is made of a set of source nodes $\{S_A, S_B, \ldots\}$, relay nodes $\{R_1, R_2, \ldots\}$ and destination nodes $\{D_1, D_2, \ldots\}$. Each source node is connected with its source message $\{b_A, b_B, \ldots\}$, which are assumed independent among source nodes. Similarly each destination node is a target for a given message.

The signal space throughout the whole network is divided into mutually orthogonal subspaces called stages. This means, that the inner product of two signals from different stages is equal zero. With the assumption of Gaussian noise, this implies that at the receiver, signals from different stages can be separated.

From the perspective of a relay, at least two stages are needed. This follows from the so-called half-duplex constraint. This is a physical limitation given by the hardware, which is not able to transmit and receive simultaneously on the same channel. Denoting the received signal at R_j as x_j and the transmitted signal as v_j, the half-duplex constraint can be expressed as $\langle x_j; v_j \rangle = 0$. Since the data transmitted by a relay are often causally connected with the data received, we will consider the half-duplex constraint to be fulfilled by a separation in the time domain. The stage where the relay receives a signal composed from multiple sources is usually called the *multiple access channel* (MAC) stage and the opposite, where the relay transmits to, possibly multiple, receiving nodes, the *broadcast channel* (BC) stage.

In Figure 2.3 we see an example network. The edges represent individual signal paths and are numbered according to the stages, in which they operate. The source data is labeled as b_A, b_B, b_C, the many to one relay functions as χ_1, χ_2 and the relay target symbols as b_1, b_2. This example nicely demonstrates how the source information travels trough the network and gets encapsulated at individual relay nodes. From this encapsulation we use the following terms:

- ***Hierarchical Network Code map*** **(HNC map)** a many to one function of the received symbols at a relay. In this case HNC maps are χ_1 and χ_2.

- ***hierarchical symbol*** the output from a HNC map at a relay. In this case b_1 is a hierarchical symbol at R_1 and b_2 a hierarchical symbol at R_2.

- ***hierarchical MAC*** **(H-MAC) stage** a MAC stage as a part of the network hierarchy. In this case H-MAC stages are stage 1 w.r.t. R_1 and stage 2 w.r.t. R_2.

- ***hierarchical BC*** **(H-BC) stage** a BC stage as a part of the network hierarchy. In this case H-BC stages are stage 2 w.r.t R_1 and stage 3 w.r.t R_2.

- ***global HNC map*** [1] is the mapping function which is seen from the perspective of a destination node. In this example D_1 receives the global HNC map $\chi_1(b_A, b_B)$ at stage 2 and the global HNC map $\chi_2(\chi_1(b_A, b_B), b_C)$ at stage 3. The two resulting symbols are put into a vector and denoted as $\vec{\chi}_i(\tilde{b})$, where i is the index of the destination node and $\tilde{b}$ the set of all source node symbols.

The main goal of the network is that every destination node has enough information to be able to determine its target symbol without ambiguity. In other words, the global HNC map at D_i has to be solvable for b_i. This holds

[1] defined for cycle free networks only

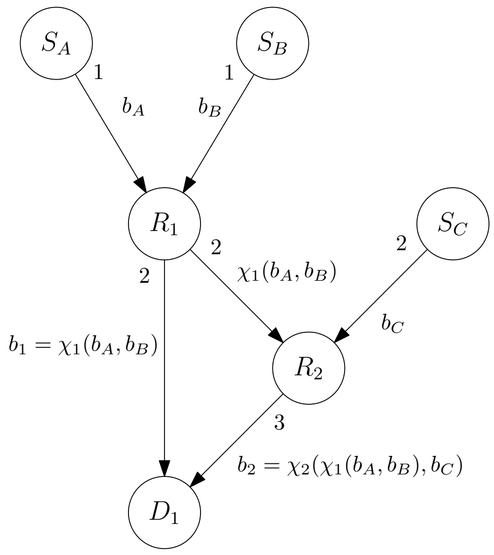


Figure 2.3: Example network.

if and only if

$$\forall \tilde{b}(b_i), \tilde{b}(b_i') : b_i \neq b_i' \Rightarrow \vec{\chi}_i \left(\tilde{b}(b_i)\right) \neq \vec{\chi}_i \left(\tilde{b}(b_i')\right), \tag{2.1}$$

where $\tilde{b}(b_i)$ is the set of all source symbols consistent with b_i.

2.1.2 Node model

In this section we discuss the inner structure and the processing chain of a node. Because source and destination nodes can be interpreted as special cases of a relay node, we can focus only on relay nodes. We consider only the case of a *layered network coded modulation* (NCM), where the forward error correction coding is decoupled from the hierarchical network structure. In this case standard single user error correction codes can be used. For a general case see [3].

A general processing chain is shown in figure 2.4. We see a simple network with two source nodes and one relay. First the two source data symbols get *forward error correction* (FEC) encoded and mapped onto constellation points s_A, s_B. Since both source nodes transmit in one stage, their constellations get combined according to the relative channel parametrization h. This way we get a hierarchical channel-combined constellation point u. As depicted in the node structure figure, the processing of the relay can be divided into three phases, the front-end, node processing and back-end. In the first phase, the received signal from the H-MAC stage x is processed into the target

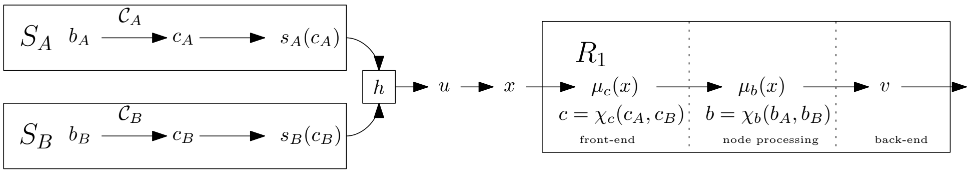


Figure 2.4: Node processing structure.

code symbol decoding metric. The metric can have different forms such as some sort of a soft information measure or a hard decision. The target code symbol c is, in general, a many to one function of the source code symbols c_A, c_B. However μ_c must be a sufficient statistic for the decoding μ_b. The target symbol measure μ_b is an information measure of the target hierarchical symbol b and is processed in the second phase. The back-end is responsible of the transmission of the hierarchical information measure μ_b in the following H-BC stage.

Chapter 3

H-MAC Estimator Derivation and Theoretical Analysis

3.1 System Model

For the purpose of this work we will consider a single 2-component H-MAC stage with two source nodes and one relay. We assume a perfect symbol-timing synchronization among all three nodes. We have two binary source messages $\mathbf{b}_A, \mathbf{b}_B \in \mathbb{F}_2^m$ of length m. At the source nodes they get encoded by a common binary linear FEC $\mathcal{C}$ with rate $R = \frac{m}{N}$. The resulting code vectors $\mathbf{c}_A = \mathcal{C}(\mathbf{b}_A), \mathbf{c}_B = \mathcal{C}(\mathbf{b}_B) \in \mathbb{F}_2^N$ are symbol-wise mapped on constellation points using the BPSK alphabet $\mathcal{A}_s = \{\pm 1\}$ as $s(c_n) = 1 - 2c_n$ at both source nodes. Both HNC maps χ_b, χ_c are realized by the XOR function. This way we get a so-called *minimal* HNC map. This means that given any two elements from $\{b_n, b_{A,n}, b_{B,n}\}$, where $b_n = \chi_B(b_{A,n}, b_{B,n})$, we can uniquely determine the third one. Further since we use a common linear code on both sources, it holds that

$$\mathbf{c} = \chi_c\left(\mathcal{C}(\mathbf{b}_A), \mathcal{C}(\mathbf{b}_B)\right) = \mathcal{C}\left(\chi_b(\mathbf{b}_A, \mathbf{b}_B)\right). \tag{3.1}$$

Together with the fact, that the FEC is decoupled from the hierarchical processing, this gives us an isomorphic layered NCM.

The 2-source H-MAC channel is assumed memoryless with AWGN modeled as

$$\begin{aligned}\mathbf{x} &= h_A\mathbf{s}(\mathcal{C}(\mathbf{b}_A)) + h_B\mathbf{s}(\mathcal{C}(\mathbf{b}_B)) + \mathbf{w}\\ &= h_A\mathbf{s}(\mathbf{c}_A) + h_B\mathbf{s}(\mathbf{c}_B) + \mathbf{w}\\ &= \mathbf{u}(\mathbf{c}_A, \mathbf{c}_B) + \mathbf{w},\end{aligned} \tag{3.2}$$

where the fading coefficients are given by magnitude and phase as

$$h_A = 1\,\mathrm{e}^{\mathrm{j}\,\phi_A}, \quad h_B = \eta\,\mathrm{e}^{\mathrm{j}\,\phi_B}, \quad 0 \leq \eta \leq 1 \tag{3.3}$$

and $\mathbf{w}$ is the AWGN with σ_w^2 variance per dimension. The frame observation is of length N and the SNR will be related w.r.t. S_A and denoted as $\gamma_x = \frac{\mathrm{E}[|s|^2]}{\sigma_w}$.

3.2 H-MAC Channel Phase Properties

In this section we will analyze the *relative* channel parametrization, which is given by mutual fading of the two source signals. It has a significant impact on the decoding process of the hierarchical data since it causes a nonlinear transformation of the resulting hierarchical constellation observed by the relay node.

3.2.1 H-MAC Channel Phase Invariance

Since the target symbol at R is a many-to-one function, multiple source symbol combinations $(s(c_{A,n}), s(c_{B,n}))$ correspond to one H-code symbol c_n at the relay. This phenomenon is called *hierarchical dispersion.* In our observation model, it demonstrates through the phase ambiguity. Our hierarchical channel-combined constellation point is given as

$$u_n(\boldsymbol{\varphi}, c_n = \chi_c(c_{A,n}, c_{B,n})) = \mathrm{e}^{\mathrm{j}\,\varphi_A}\, s(c_{A,n}) + \eta\,\mathrm{e}^{\mathrm{j}\,\varphi_B}\, s(c_{B,n}), \tag{3.4}$$

where $\boldsymbol{\varphi} = [\varphi_A, \varphi_B]^T$ is the parametric channel phase vector. Using the constellation mappers $s(c) = 2c - 1$ and the fact, that our HNC map fulfills the property $\chi_c(c_{A,n}, c_{B,n}) = \chi_c(1 - c_{A,n}, 1 - c_{B,n})$, it is easy to show that the following holds

$$u_n(\boldsymbol{\varphi}, c_n) = u_n(\boldsymbol{\varphi} + [(2k_1 + 1)\pi, (2k_2 + 1)\pi]^T, \chi_c(1 - c_{A,n}, 1 - c_{B,n})) \tag{3.5}$$

for $k_1, k_2 \in \mathbb{Z}$. In other words, a shift by a odd multiple of π in either coordinate of $\boldsymbol{\varphi}$ does not change the hierarchical code symbol c_n.

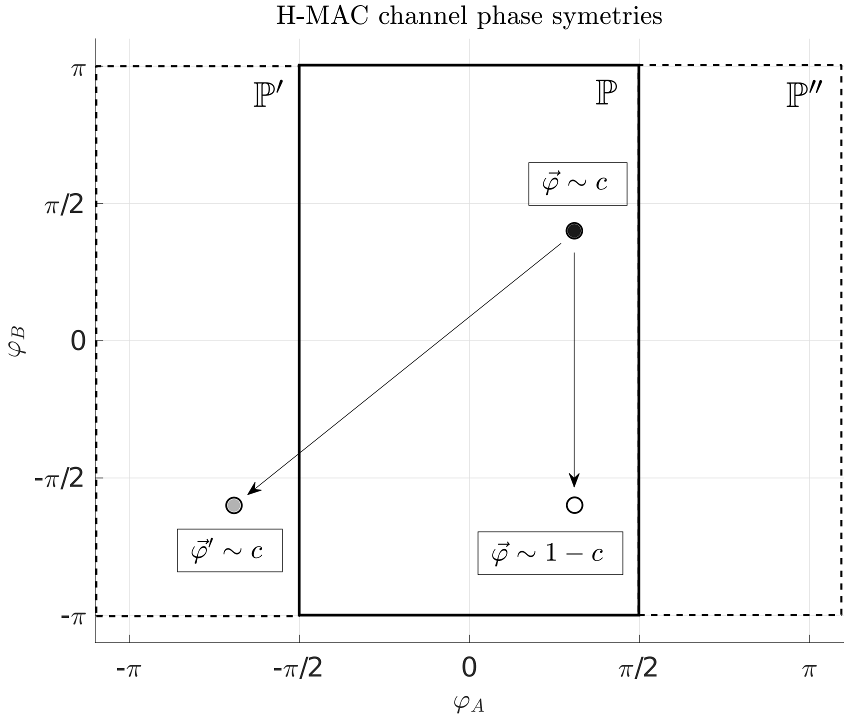


Figure 3.1: H-MAC channel phase symmetries. Solid line marks $\mathbb{P}$, dashed line equivalent shifted versions $\mathbb{P}', \mathbb{P}''$. The full circle is the true phase vector for target symbol c, the gray filled circle is an equivalent phase vector w.r.t. c and the empty circle is the phase vector corresponding to the a flipped target symbol $1 - c$.

This means, that from the perspective of the relay we can define a rectangle

$$\mathbb{P} = \left\{ [\varphi_A, \varphi_B] : -\frac{\pi}{2} < \varphi_A \leq \frac{\pi}{2}, -\pi < \varphi_B \leq \pi \right\}, \tag{3.6}$$

for which it holds that phase vectors $\boldsymbol{\varphi} \in \mathbb{P}$ are unambiguous w.r.t. c. From the perspective of a H-data aided CSE, all phase estimates $\boldsymbol{\varphi} \notin \mathbb{P}$ can be shifted according to Equation 3.5 to an equivalent solution laying in $\mathbb{P}$. Notice that this does not generally hold for a classical full data c_A, c_B aided estimator. In Figure 3.1 we see a phase vector $\in \mathbb{P}$ marked with a black filled circle and its equivalent shifted version marked with a gray filled circle.

3.2.2 H-MAC H-symbol Dependence on Channel Phase

From the HNC map it follows that for the target hierarchical symbol c_n to change, exactly one source symbol has to change. This change can be translated to an equivalent change in the channel parametrization as it follows from Equation 3.4. It is easy to see, that a flip of the target hierarchical symbol c_n is connected with a sign change of the corresponding constellation point, which in turn relates to a phase change of an odd multiple of π in the respective coordinate of $\boldsymbol{\varphi}$. Thus we can write

$$\begin{aligned} u_n(\boldsymbol{\varphi}, c) &= u_n(\boldsymbol{\varphi} + (2k+1)\pi[1,0]^T, 1 - c_n) \\ u_n(\boldsymbol{\varphi}, c) &= u_n(\boldsymbol{\varphi} + (2k+1)\pi[0,1]^T, 1 - c_n) \end{aligned} \tag{3.7}$$

for $k \in \mathbb{Z}$. Figure 3.1 illustrates a shift in the coordinate φ_B.

3.3 Hierarchical Soft Output Demodulator (H-SODEM)

In this section we will derive a hierarchical demodulator considering our particular system model, a general case can be found in [3]. From the perspective of the node processing chain of R, the SODEM implements a part of the front-end. It processes the received signal $\mathbf{x}$ and outputs the soft decoding metric μ_c, which is in our case given by the log likelihood ratio. Because our channel is memoryless, it holds that $p(\mathbf{x}|\mathbf{c}_A, \mathbf{c}_B) = \prod_n p(x_n|c_{A,n}, c_{B,n})$ and thus the likelihood can be evaluated symbol-wise. Since we are interested only in the hierarchical symbol c_n, we marginalize the likelihood over all combinations of $c_{A,n}, c_{B,n}$ consistent with it, denoted as $\tilde{c}_n : \chi_c(\tilde{c}_n) = c_n$. The marginalized likelihood can be expressed as

$$p(x_n|c_n) = \frac{\sum_{\tilde{c}_n:\chi_c(\tilde{c}_n)=c_n} p(x_n|\tilde{c}_n)p(\tilde{c}_n)}{\sum_{\tilde{c}_n:\chi_c(\tilde{c}_n)=c_n} p(\tilde{c}_n)}. \tag{3.8}$$

Assuming that $c_{A,n}, c_{B,n} \in \{0,1\}$ are equiprobable, we get

$$\begin{aligned} p(c_{i,n}) &= \frac{1}{2} \\ p(\tilde{c}_n) &= \frac{1}{2^2} \\ p(c_n) &= \sum_{\tilde{c}_n:\chi_c(\tilde{c}_n)=c_n} p(\tilde{c}_n) = 2p(\tilde{c}_n) = \frac{1}{2}, \end{aligned} \tag{3.9}$$

where we used the fact that our HNC map is minimal and thus one c_n corresponds to exactly two different combinations $\tilde{c}_n$. After substitution of Equations 3.9 into Equation 3.8 we get

$$p(x_n|c_n) = \frac{1}{2} \sum_{\tilde{c}_n:\chi_c(\tilde{c}_n)=c_n} p(x_n|\tilde{c}_n). \tag{3.10}$$

With the assumption of a known the channel parametrization h_A, h_B at the demodulator, we can substitute the channel model given in Equation 3.2 and get

$$\begin{aligned} p(x_n|c_n) &= \frac{1}{2} \sum_{\tilde{c}_n : \chi_c(\tilde{c}_n) = c_n} p_w(x_n - u_n(\tilde{c}_n)) \\ &= \frac{1}{2\pi\sigma_w^2} \sum_{\tilde{c}_n : \chi_c(\tilde{c}_n) = c_n} \exp\left(-\frac{1}{\sigma_w^2} \left| x_n - u_n(\tilde{c}_n) \right|^2 \right). \end{aligned} \tag{3.11}$$

The final hierarchical decoding metric μ_c is given as s symbol-wise LLR

$$\mathrm{LLR}_n = \log\left(\frac{p(x_n|c_n = 0)}{p(x_n|c_n = 1)}\right). \tag{3.12}$$

3.4 Hierarchical MAC Channel Phase Estimator

Now we derive a H-Data decision aided H-MAC channel *maximum likelihood* (ML) phase estimator based on [2]. This means the estimator does not have the individual messages $\mathbf{b}_A, \mathbf{b}_B$ nor their coded versions $\mathbf{c}_A, \mathbf{c}_B$ available. It has only the many-to-one hierarchical function estimate $\hat{\mathbf{c}} = \chi_c(\mathbf{c}_A, \mathbf{c}_B)$ given from the H-decoder. We assume that the estimate is reliable $\hat{\mathbf{c}} = \mathbf{c}$ and both the noise variance σ_w^2 and η are known.

3.4.1 Estimator Derivation

Using the fact of the memoryless channel, we can proceed symbol-wise. We start with the observation likelihood

$$p(x|\varphi_A, \varphi_B, c_A, c_B) \equiv \exp\left(-\frac{1}{\sigma_w^2} \left| x_n - \mathrm{e}^{\mathrm{j}\,\varphi_A} s(c_{A,n}) - \eta\, \mathrm{e}^{\mathrm{j}\,\varphi_B} s(c_{B,n}) \right|^2 \right), \tag{3.13}$$

where we dropped unimportant scaling. Since the constellation mappers are one-to-one functions, we introduce $s_{A,n} = s(c_{A,n})$, and $s_{B,n} = s(c_{B,n})$. After some manipulations we get

$$\begin{aligned} &\left| x_n - \mathrm{e}^{\mathrm{j}\,\varphi_A} s_{A,n} - \eta\, \mathrm{e}^{\mathrm{j}\,\varphi_B} s_{B,n} \right|^2 = |x|^2 + \left| \mathrm{e}^{\mathrm{j}\,\varphi_A} s_{A,n} \right|^2 + \left| \eta\, \mathrm{e}^{\mathrm{j}\,\varphi_B} s_{B,n} \right|^2 \\ &+ 2\Re\left\{ \mathrm{e}^{\mathrm{j}(\varphi_A - \varphi_B)} \eta s_{A,n} s_{B,n}^* \right\} - 2\Re\left\{ x\, \mathrm{e}^{-\mathrm{j}\,\varphi_A} s_{A,n}^* \right\} - 2\Re\left\{ x\, \mathrm{e}^{-\mathrm{j}\,\varphi_B} \eta s_{B,n}^* \right\}. \end{aligned} \tag{3.14}$$

For BPSK we have $|s_{A,n}| = |s_{B,n}| = 1$, then it follows that only the last three terms are functions of $\varphi_A, \varphi_B, s_{A,n}, s_{B,n}$. The rest can be neglected because it acts only as a multiplicative constant w.r.t. the likelihood. When applied and substituted back into Equation 3.13 we obtain

$$p(x|\varphi_A,\varphi_B,s_{A,n},s_{B,n}) \equiv \exp\left(-\frac{2}{\sigma_w^2}\eta\Re\left\{\mathrm{e}^{\mathrm{j}(\varphi_A-\varphi_B)}\,s_{A,n}s^*_{B,n}\right\}\right) \\ \exp\left(\frac{2}{\sigma_w^2}\left(\Re\left\{x\,\mathrm{e}^{-\mathrm{j}\,\varphi_A}\,s^*_{A,n}\right\}+\eta\Re\left\{x\,\mathrm{e}^{-\mathrm{j}\,\varphi_B}\,s^*_{B,n}\right\}\right)\right). \tag{3.15}$$

Still there is a dependence on the individual source constellation points $s_{A,n}, s_{B,n}$. We can use the minimal property of the HNC map, giving us $s_{B,n}$ as a function of the hierarchical symbol and $s_{A,n}$

$$s_{B,n} = s_B(s_{A,n},c_n) = 1-2c_{B,n} = 1-2(c_n\oplus c_{A,n}) = (1-2c)s_{A,n}. \tag{3.16}$$

Which gives $s_{B,n} = s_{A,n}$ for $c_n = 1$ and $s_{B,n} = -s_{A,n}$ otherwise. Combined with the assumption of IID source symbols, which gives $p(s_{A,n}) = p(s_{B,n}) = \frac{1}{2} = p(c_n)$, the marginalization can be written as

$$\begin{aligned} p(x|\varphi_A,\varphi_B,c_n) &\equiv \frac{1}{p(c_n)}\sum_{(s_{A,n},s_{B,n}):c_n} p(x|\varphi_A,\varphi_B,s_{A,n},s_{B,n})p(s_{A,n})p(s_{B,n}) \\ &= \frac{1}{2}\sum_{s_{A,n}} p(x|\varphi_A,\varphi_B,s_{A,n},s_B(s_{A,n},c_n)). \end{aligned} \tag{3.17}$$

We may notice that our 2-source BPSK NCM with XOR HNC map results in $s_{A,n}s^*_{B,n}$ being a one-to-one function of c_n namely

$$s_{A,n}s^*_{B,n} = s(c_n) = (1-2c_n)|s_{A,n}|^2 = 1-2c_n. \tag{3.18}$$

Using this together with $\mathcal{A}_s = \{\pm 1\}$, the marginalized H-symbol conditioned likelihood (dropping unimportant scaling) yields

$$\begin{aligned} p(x|\varphi_A,\varphi_B,c_n) \equiv \sum_{s_{A,n}\in\mathcal{A}_s} & \exp\left(-\frac{2}{\sigma_w^2}\eta\Re\left\{\mathrm{e}^{\mathrm{j}(\varphi_A-\varphi_B)}\,s(c_n)\right\}\right) \\ & \exp\left(\frac{2}{\sigma_w^2}\left(\Re\left\{x\,\mathrm{e}^{-\mathrm{j}\,\varphi_A}\,s^*_{A,n}\right\}+\eta\Re\left\{x\,\mathrm{e}^{-\mathrm{j}\,\varphi_B}\,s(c_n)s^*_{A,n}\right\}\right)\right) \\ = & \exp\left(-\frac{2}{\sigma_w^2}\eta\Re\left\{\mathrm{e}^{\mathrm{j}(\varphi_A-\varphi_B)}\,s(c_n)\right\}\right) \\ & 2\cosh\left(\frac{2}{\sigma_w^2}\left(\Re\left\{x\,\mathrm{e}^{-\mathrm{j}\,\varphi_A}\right\}+\eta\Re\left\{x\,\mathrm{e}^{-\mathrm{j}\,\varphi_B}\,s(c_n)\right\}\right)\right). \end{aligned} \tag{3.19}$$

The final symbol-wise estimator metric ρ is given as a logarithm of the scaled likelihood as

$$\begin{aligned} \rho(\varphi_A,\varphi_B,c_n) = & -\frac{2}{\sigma_w^2}\eta s(c_n)\cos(\varphi_A-\varphi_B)+ \\ & \ln\cosh\left(\frac{2}{\sigma_w^2}\left(\Re\left\{x\,\mathrm{e}^{-\mathrm{j}\,\varphi_A}\right\}+\eta s(c_n)\Re\left\{x\,\mathrm{e}^{-\mathrm{j}\,\varphi_B}\right\}\right)\right) \end{aligned} \tag{3.20}$$

and for the whole frame as a simple sum

$$\rho_N(\varphi_A, \varphi_B, \mathbf{c}) = \sum_{n=1}^{N} \rho(\varphi_A, \varphi_B, c_n). \tag{3.21}$$

The estimate is then given such that it maximizes the derived metric

$$\hat{\boldsymbol{\varphi}} = \arg \max_{\varphi_A, \varphi_B} \rho_N(\varphi_A, \varphi_B, \mathbf{c}) \tag{3.22}$$

3.4.2 Estimator Metric Approximation

We are interested in the number and position of the estimator metric maxima, however the exact form given in Equation 3.21 is difficult to analyze. In this section we derive an analytically tractable approximation.

The fully expanded exact expression for the metric is

$$\begin{aligned}\rho_N(\boldsymbol{\varphi}, \mathbf{c}) = &\sum_{n=1}^{N} -\frac{2}{\sigma_w^2}\eta s(c_n)\cos(\varphi_A - \varphi_B) + \\ &\sum_{n=1}^{N} \ln\cosh\left(\frac{2}{\sigma_w^2}\left(\Re\left\{x\,\mathrm{e}^{-\mathrm{j}\varphi_A}\right\} + \eta s(c_n)\Re\left\{x\,\mathrm{e}^{-\mathrm{j}\varphi_B}\right\}\right)\right).\end{aligned} \tag{3.23}$$

If we assume a very long frame $N \to \infty$ and equiprobable target symbols $\Pr(c_n = 1) = \Pr(c_n = -1)$. In practice, the first assumption is rather fulfilled since frame lengths of today used LDPC codes are in the order of 64800. The equiprobability of the hierarchical symbol values directly follows from the minimal HNC map. Apparently the first sum goes to zero

$$-\frac{2}{\sigma_w^2}\cos(\varphi_A - \varphi_B)\sum_{n=1}^{N} s(c_n) \to 0, \tag{3.24}$$

while the second is a monotonically increasing function of N since

$$\ln(\cosh(x)) \geq 0, \quad \forall x. \tag{3.25}$$

With a further assumption of the operation in the high-SNR regime, $\frac{2}{\sigma_w^2} \gg 1$, we can make use of the following approximation

$$\ln\left(\cosh(x)\right) = \ln\left(\frac{1}{2}\right) + \ln(\mathrm{e}^{x} + \mathrm{e}^{-x}) \approx \ln\left(\frac{1}{2}\right) + |x| \text{ for } x \gg 1. \tag{3.26}$$

Both the additive $\ln(\frac{1}{2})$ and multiplicative $\frac{2}{\sigma_w^2}$ do not influence the positions of extrema so we can drop them and obtain

$$\rho_N'(\boldsymbol{\varphi}, \mathbf{c}) \approx \sum_{n=1}^{N} \left|\Re\left\{x_n\,\mathrm{e}^{-\mathrm{j}\varphi_A}\right\} + \eta s(c_n)\Re\left\{x_n\,\mathrm{e}^{-\mathrm{j}\varphi_B}\right\}\right|. \tag{3.27}$$

From the perspective of an off-line analysis, we can substitute the observation model with the true phase vector $\boldsymbol{\phi} = [\phi_A, \phi_B]^T$ (Equation 3.2) for x_n. After some manipulations we get

$$\begin{aligned}
\rho'_N \approx \sum_{n=1}^{N} \Big| &\Big(s_{A,n} \cos(\phi_A - \varphi_A) + \eta s_{B,n} \cos(\phi_B - \varphi_A) + \Re\{w_n\} \cos(\varphi_A) \\
&+ \Im\{w_n\} \sin(\varphi_A) \Big) \\
&+ \eta s(c_n) \Big(s_{A,n} \cos(\phi_A - \varphi_B) + \eta s_{B,n} \cos(\phi_B - \varphi_B) + \Re\{w_n\} \cos(\varphi_B) \\
&+ \Im\{w_n\} \sin(\varphi_B) \Big) \Big| \\
= \sum_{n=1}^{N} \Big| & s_{A,n} \Big(\cos(\phi_A - \varphi_A) + \eta s(c_n) \cos(\phi_A - \varphi_B) \Big) \\
&+ \eta s_{B,n} \Big(\cos(\phi_B - \varphi_A) + \eta s(c_n) \cos(\varphi_B - \varphi_B) \Big) \\
&+ \Re\{w_n\} \Big(\cos(\varphi_A) + \eta s(c_n) \cos(\varphi_B) \Big) \\
&+ \Im\{w_n\} \Big(\sin(\varphi_A) + \eta s(c_n) \sin(\varphi_B) \Big) \Big| .
\end{aligned} \tag{3.28}$$

At this point we make an assumption of a noiseless channel $|\Re\{w_n\}|, |\Im\{w_n\}| \to 0$. Since $s_{A,n} \in \{\pm 1\}$, the modulus argument of each n-th term in Equation 3.28 can be multiplied by $s_{A,n}$ not changing the overall sum value. Then we can write

$$\begin{aligned}
\rho'_N(\boldsymbol{\varphi}, \mathbf{c}) \approx \sum_{n=1}^{N} \Big| & \cos(\phi_A - \varphi_A) + \eta s(c_n) \cos(\phi_A - \varphi_B) \\
&+ \eta s(c_n) \cos(\phi_B - \varphi_A) + \eta^2 \cos(\phi_B - \varphi_B) \Big| .
\end{aligned} \tag{3.29}$$

Further we notice that c_n is the only part that depends on the summation index. That is as expected, realizing that the relay metric is marginalized over the hierarchical dispersion. Using the assumption of c_n having equiprobable values and N being large, the sum can be reordered and split in half according to the value of $s(c_n) \in \{\pm 1\}$. Dropping the multiplicative $\frac{N}{2}$ we get the final expression

$$\rho'(\boldsymbol{\varphi}) = \frac{2}{N} \rho'_N(\boldsymbol{\varphi}, \mathbf{c}) = \rho'(\boldsymbol{\varphi}, 1) + \rho'(\boldsymbol{\varphi}, -1) \tag{3.30}$$

where

$$\begin{aligned}
\rho'(\boldsymbol{\varphi}, s) = \Big| & \cos(\phi_A - \varphi_A) + \eta s \cos(\phi_A - \varphi_B) \\
&+ \eta s \cos(\phi_B - \varphi_A) + \eta^2 \cos(\phi_B - \varphi_B) \Big| .
\end{aligned} \tag{3.31}$$

3.4.3 Location of the Metric Extrema

In this section we use the previously derived approximation and analyze its extrema, focusing on the number and position of maxima. We will proceed separately for each of the four sign combination of the moduli arguments $\rho'(\boldsymbol{\varphi}, 1), \rho'(\boldsymbol{\varphi}, -1)$ from Equation 3.30.

[+,+]

If the moduli arguments of both $\rho'(\boldsymbol{\varphi}, 1)$ and $\rho'(\boldsymbol{\varphi}, -1)$ are positive, we can remove the moduli and after simplification we get

$$\rho'_{++}(\boldsymbol{\phi}) = 2\cos(\phi_A - \varphi_A) + 2\eta^2 \cos(\phi_B - \varphi_B). \tag{3.32}$$

Performing the first and second derivative w.r.t. $\boldsymbol{\varphi}$ it yields

$$\begin{aligned}
\nabla_{\varphi} &= \begin{bmatrix} \dfrac{\partial \rho'_{++}}{\partial \varphi_A}, \dfrac{\partial \rho'_{++}}{\partial \varphi_B} \end{bmatrix}^T = [2\sin(\phi_A - \varphi_A),\, \eta^2 \sin(\phi_B - \varphi_B)]^T \\
\mathbf{H}_{\varphi} &= \begin{bmatrix} \frac{\partial^2 \rho'_{++}}{\partial \varphi_A^2} & \frac{\partial^2 \rho'_{++}}{\partial \varphi_A \partial \varphi_B} \\ \frac{\partial^2 \rho'_{++}}{\partial \varphi_A \partial \varphi_B} & \frac{\partial^2 \rho'_{++}}{\partial \varphi_B^2} \end{bmatrix} = \begin{bmatrix} -2\cos(\phi_A - \varphi_A) & 0 \\ 0 & -2\eta^2 \cos(\phi_B - \varphi_B) \end{bmatrix}.
\end{aligned} \tag{3.33}$$

Clearly, we see that both $\boldsymbol{\varphi}_0 = [\phi_A, \phi_B]^T, \boldsymbol{\varphi}_1 = [\phi_A, \phi_B + \pi]^T$ zero the gradient. From the metric periodicity we get an infinity of other zero gradient points, but in terms of Section 3.2.1 they are equivalent to $\boldsymbol{\varphi}_0$ and $\boldsymbol{\varphi}_1$. Evaluating the definiteness of $\mathbf{H}_{\varphi}$ at $\boldsymbol{\varphi}_0$ and $\boldsymbol{\varphi}_1$ we get

$$\begin{aligned}
\mathbf{H}_{\varphi}\Big|_{\boldsymbol{\varphi}_0} &= \begin{bmatrix} -2 & 0 \\ 0 & -2\eta^2 \end{bmatrix} \Rightarrow \text{negative definite} \\
\mathbf{H}_{\varphi}\Big|_{\boldsymbol{\varphi}_1} &= \begin{bmatrix} -2 & 0 \\ 0 & 2\eta^2 \end{bmatrix} \Rightarrow \text{indefinite}
\end{aligned} \tag{3.34}$$

thus at $\boldsymbol{\varphi}_0$ the metric exhibits a maximum and at $\boldsymbol{\varphi}_1$ there is a saddle point. Now we need to check if the assumption of positive moduli arguments stated at the beginning of this section is met at both $\boldsymbol{\varphi}_0$ and $\boldsymbol{\varphi}_1$. Doing the substitution we get the following conditions

$$\begin{aligned}
\eta^2 \pm 2\eta \cos(\phi_A - \phi_B) + 1 \geq 0 \text{ for } \boldsymbol{\varphi}_0 \\
1 - \eta^2 \geq 0 \text{ for } \boldsymbol{\varphi}_1
\end{aligned} \tag{3.35}$$

Since $\eta^2 \pm 2\eta \cos(\phi_A - \phi_B) + 1 \geq \eta^2 - 2\eta + 1 = (\eta - 1)^2 \geq 0$ and $0 < \eta \leq 1$, both conditions are always fulfilled. This proves that the metric will always have a maximum at the true phase vector $\boldsymbol{\varphi}_0 = \boldsymbol{\phi}$.

[+,−]

We will proceed similarly as in the previous section. After replacing the moduli for this case we get

$$\rho'_{+-}(\boldsymbol{\varphi}) = 2\eta(\cos(\phi_B - \varphi_A) + \cos(\phi_A - \varphi_B)). \tag{3.36}$$

The derivatives work out to

$$\begin{aligned}\nabla_{\varphi} &= [2\eta\sin(\phi_B - \varphi_A),\, 2\eta\sin(\phi_A - \varphi_B)]^T \\ \mathbf{H}_{\varphi} &= \begin{bmatrix} -2\eta\cos(\phi_B - \varphi_A) & 0 \\ 0 & -2\eta\cos(\phi_A - \varphi_B) \end{bmatrix}.\end{aligned} \tag{3.37}$$

Again, we get two distinct stationary points $\boldsymbol{\varphi}_0 = [\phi_B, \phi_A]^T, \boldsymbol{\varphi}_1 = [\phi_B, \phi_A + \pi]^T$. From

$$\begin{aligned}\mathbf{H}_{\varphi}\Big|_{\boldsymbol{\varphi}_0} &= \begin{bmatrix} -2\eta & 0 \\ 0 & -2\eta \end{bmatrix} \Rightarrow \text{negative definite} \\ \mathbf{H}_{\varphi}\Big|_{\boldsymbol{\varphi}_1} &= \begin{bmatrix} -2\eta & 0 \\ 0 & 2\eta \end{bmatrix} \Rightarrow \text{indefinite}\end{aligned} \tag{3.38}$$

we get another maximum at $\boldsymbol{\varphi}_0$ and a saddle point at $\boldsymbol{\varphi}_1$. The sign conditions follow as

$$\begin{aligned}(\eta^2 + 1)\cos(\phi_A - \phi_B) + 2\eta &\geq 0 \text{ and} \\ (\eta^2 + 1)\cos(\phi_A - \phi_B) - 2\eta &\leq 0 \text{ for } \boldsymbol{\varphi}_0\end{aligned} \tag{3.39}$$

$$(1 - \eta^2)\cos(\phi_A - \phi_B) = 0 \text{ for } \boldsymbol{\varphi}_1. \tag{3.40}$$

In this case neither of them is trivially fulfilled. The filled area in Figure 3.2 contains all points which meet both conditions for $\boldsymbol{\varphi}_0$ and thus the metric contains a false maximum at $[\boldsymbol{\phi}_B, \boldsymbol{\phi}_A]^T$.

[−,−],[−,+]

In the cases of the two remaining sign combinations, we could proceed analogously, however the derivation would be very similar with opposite signs of the gradients and hessian matrices and would not lead to any new maxima or saddle points.

3.4.4 Values of Metric Maxima

In the previous section we concluded that there are in general two maxima. Now we are interested in their absolute values and what is more important,

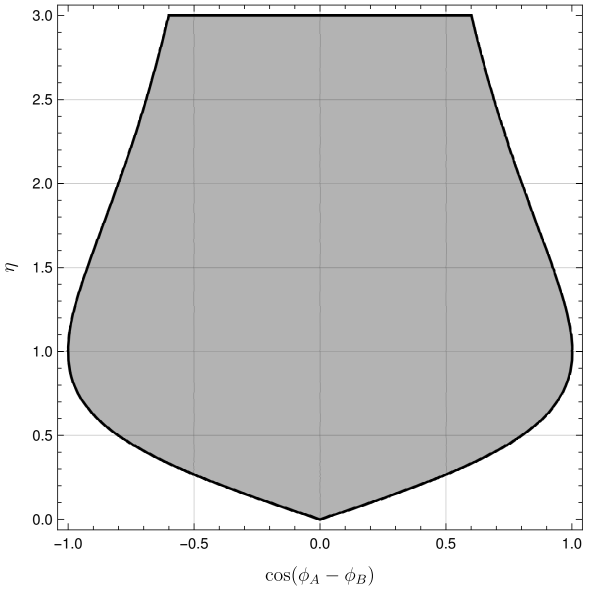


Figure 3.2: Area for which the estimator metric exhibits a false maximum at $[\boldsymbol{\phi}_B, \boldsymbol{\phi}_A]^T$.

their mutual relation. We start with Equation 3.30, fix the absolute error introduced by dropping constants and obtain

$$\rho_N(\boldsymbol{\varphi}) \approx \ln(\frac{1}{2}) + \frac{N}{2}\frac{2}{\sigma_w^2}\Big(\rho'(\boldsymbol{\varphi}, 1) + \rho'(\boldsymbol{\varphi}, -1)\Big). \tag{3.41}$$

An evaluation at the true phase vector yields

$$\rho_N(\boldsymbol{\varphi})\Big|_{\boldsymbol{\varphi}=[\phi_A,\phi_B]^T} \approx N\ln\left(\frac{1}{2}\right) + \frac{2N}{\sigma_w^2}(1+\eta^2), \tag{3.42}$$

at the false phase vector the metric evaluates to

$$\rho_N(\boldsymbol{\varphi})\Big|_{\boldsymbol{\varphi}=[\phi_B,\phi_A]^T} \approx N\ln\left(\frac{1}{2}\right) + \frac{2N}{\sigma_w^2}(2\eta). \tag{3.43}$$

For the difference we can write

$$\rho_N([\phi_A,\phi_B]^T) - \rho_N([\phi_B,\phi_A]^T) \approx \frac{2N}{\sigma_w^2}(1+\eta^2-2\eta) = \frac{2N}{\sigma_w^2}(1-\eta)^2 \geq 0. \tag{3.44}$$

We see that the maximum at the true phase is always superior except for $\eta = 1$, in which case, they have equal values. From Figure 3.2 it follows that for this particular case, the second maximum exists independently on the value of $\cos(\phi_A - \phi_B)$. However when looking at the hierarchical channel combined observation given in Equation 3.4, for $\eta = 1$, it holds true that

$$u_n([\phi_A,\phi_B]^T, c_n) = u_n([\phi_B,\phi_A]^T, c_n),\ \forall c_n \tag{3.45}$$

since

$$\mathrm{e}^{\mathrm{j}\,\phi_A}\, s(c_{A,n}) + \mathrm{e}^{\mathrm{j}\,\phi_B}\, s(c_{B,n}) = \mathrm{e}^{\mathrm{j}\,\phi_B}\, s(1-c_{A,n}) + \mathrm{e}^{\mathrm{j}\,\phi_A}\, s(1-c_{B,n}) \tag{3.46}$$

for $c_{A,n} \neq c_{B,n}$ and

$$\mathrm{e}^{\mathrm{j}\,\phi_A}\,s(c_{A,n}) + \mathrm{e}^{\mathrm{j}\,\phi_B}\,s(c_{B,n}) = \mathrm{e}^{\mathrm{j}\,\phi_B}\,s(c_{A,n}) + \mathrm{e}^{\mathrm{j}\,\phi_A}\,s(c_{B,n}) \tag{3.47}$$

for $c_{A,n} = c_{B,n}$. The hierarchical observation is invariant w.r.t. the phase vector flip because the target hierarchical symbol c_n is unaffected.

As a conclusion we can state that for $\eta \neq 1$ the metric attains a global maximum at the true phase vector and when conditions (3.39) are met, a local maximum at the flipped phase vector. In the singular case when $\eta = 1$, there are two equally valued local maxima but in this situation they both yield the same observation and are equivalent from the perspective of the relay.

3.4.5 Cramer-Rao Lower Bound Analysis

The Cramer-Rao Lower Bound (CRLB) is a tool by means of which one can establish a lower bound on the estimate variance of any unbiased estimator. For details see [1]. The lower bound is conditioned by the regularity condition which is in our case given as

$$\mathrm{E}\left[\frac{\partial \ln(p(\mathbf{x}|\boldsymbol{\varphi},\mathbf{c}))}{\partial \boldsymbol{\varphi}}\right] = \mathbf{0}, \tag{3.48}$$

where E is the expectation operator. From the memoryless channel and the linearity of E follows that

$$\mathrm{E}\Big[\ln(p(\mathbf{x}|\boldsymbol{\varphi},\mathbf{c}))\Big] = \mathrm{E}\left[\sum_{n=1}^{N}\ln(p(x_n|\boldsymbol{\varphi},c_n))\right] = N\,\mathrm{E}_{x,c}\Big[p(x|\boldsymbol{\varphi},c)\Big]. \tag{3.49}$$

Realizing this we can rewrite the regularity condition as

$$\mathrm{E}\left[\frac{\partial \ln(p(x|\boldsymbol{\varphi},c))}{\partial \boldsymbol{\varphi}}\right] = \mathbf{0}. \tag{3.50}$$

Performing the derivative w.r.t. φ_A we get

$$\begin{aligned}\frac{\partial \ln(p(x|\boldsymbol{\varphi},c))}{\partial \varphi_A} =& \frac{2}{\sigma_w^2}\eta s(c)\sin(\varphi_A-\varphi_B) + \frac{2}{\sigma_w^2}\frac{\partial \Re\left\{x\,\mathrm{e}^{-\mathrm{j}\,\varphi_A}\right\}}{\partial \varphi_A}\\ &\frac{\sinh\left(\frac{2}{\sigma_w^2}\left(\Re\left\{x\,\mathrm{e}^{-\mathrm{j}\,\varphi_A}\right\}+\eta s(c)\Re\left\{x\,\mathrm{e}^{-\mathrm{j}\,\varphi_B}\right\}\right)\right)}{\cosh\left(\frac{2}{\sigma_w^2}\left(\Re\left\{x\,\mathrm{e}^{-\mathrm{j}\,\varphi_A}\right\}+\eta s(c)\Re\left\{x\,\mathrm{e}^{-\mathrm{j}\,\varphi_B}\right\}\right)\right)}\\ =& \frac{2}{\sigma_w^2}\eta s(c)\sin(\varphi_A-\varphi_B) + \frac{2}{\sigma_w^2}\Im\left\{x\,\mathrm{e}^{-\mathrm{j}\,\varphi_A}\right\}\\ &\tanh\left(\frac{2}{\sigma_w^2}\left(\Re\left\{x\,\mathrm{e}^{-\mathrm{j}\,\varphi_A}\right\}+\eta s(c)\Re\left\{x\,\mathrm{e}^{-\mathrm{j}\,\varphi_B}\right\}\right)\right).\end{aligned} \tag{3.51}$$

Similarly for φ_B

$$\begin{aligned}\frac{\partial \ln(p(x|\boldsymbol{\varphi}, c))}{\partial \varphi_B} =& \frac{2}{\sigma_w^2}\eta s(c)\sin(\varphi_B - \varphi_A) + \frac{2}{\sigma_w^2}\eta\Im\left\{x\,\mathrm{e}^{-\,\mathrm{j}\,\varphi_B}\right\} \\ & \tanh\left(\frac{2}{\sigma_w^2}\left(\eta\Re\left\{x\,\mathrm{e}^{-\,\mathrm{j}\,\varphi_B}\right\} + s(c)\Re\left\{x\,\mathrm{e}^{-\,\mathrm{j}\,\varphi_A}\right\}\right)\right).\end{aligned} \tag{3.52}$$

After substitution the regularity condition w.r.t φ_A evaluates to

$$\begin{aligned}0 =& \frac{2}{\sigma_w^2}\eta\sin(\varphi_A - \varphi_B)\,\mathrm{E}\left[s(c)\right] + \\ & \frac{2}{\sigma_w^2}\eta\,\mathrm{E}\left[\Im\left\{x\,\mathrm{e}^{-\,\mathrm{j}\,\varphi_A}\right\}\tanh\left(\frac{2}{\sigma_w^2}\left(\Re\left\{x\,\mathrm{e}^{-\,\mathrm{j}\,\varphi_A}\right\} + \eta s(c)\Re\left\{x\,\mathrm{e}^{-\,\mathrm{j}\,\varphi_B}\right\}\right)\right)\right].\end{aligned} \tag{3.53}$$

From the equiprobability of c, the first term evaluates right to zero. The second term is analytically not manageable, but a numerical Monte Carlo simulation showed that it also evaluates to 0. Similarly, the regularity condition for φ_B is fulfilled as well.

The lower bound on the individual estimate components is given by the diagonal elements of the inverse Fisher information matrix

$$\mathbf{J} = -N\,\mathrm{E}\begin{bmatrix}\frac{\partial^2 \ln(p(x|\boldsymbol{\varphi},c))}{\partial \varphi_A^2} & \frac{\partial^2 \ln(p(x|\boldsymbol{\varphi},c))}{\partial \varphi_A \partial \varphi_B} \\ \frac{\partial^2 \ln(p(x|\boldsymbol{\varphi},c))}{\partial \varphi_A \partial \varphi_B} & \frac{\partial^2 \ln(p(x|\boldsymbol{\varphi},c))}{\partial \varphi_B^2}\end{bmatrix}. \tag{3.54}$$

The lower bounds are given as

$$\begin{aligned}\mathrm{var}(\hat{\varphi}_A) &\geq [J^{-1}]_{1,1} \\ \mathrm{var}(\hat{\varphi}_B) &\geq [J^{-1}]_{2,2}.\end{aligned} \tag{3.55}$$

The second derivatives needed for the Fisher matrix computation work out to

$$\begin{aligned}\frac{\partial^2 \ln(p(x|\boldsymbol{\varphi}, c))}{\partial \varphi_A^2} =& \frac{2}{\sigma_w^2}\eta s(c)\cos(\varphi_A - \varphi_B) - \frac{2}{\sigma_w^2}\Re\left\{x\,\mathrm{e}^{-\,\mathrm{j}\,\varphi_A}\right\} \\ & \tanh\left(\frac{2}{\sigma_w^2}\left(\Re\left\{x\,\mathrm{e}^{-\,\mathrm{j}\,\varphi_A}\right\} + \eta s(c)\Re\left\{x\,\mathrm{e}^{-\,\mathrm{j}\,\varphi_B}\right\}\right)\right) + \\ & \left[\frac{\frac{2}{\sigma_w^2}\Im\left\{x\,\mathrm{e}^{-\,\mathrm{j}\,\varphi_A}\right\}}{\cosh\left(\frac{2}{\sigma_w^2}\left(\Re\left\{x\,\mathrm{e}^{-\,\mathrm{j}\,\varphi_A}\right\} + \eta s(c)\Re\left\{x\,\mathrm{e}^{-\,\mathrm{j}\,\varphi_B}\right\}\right)\right)}\right]^2\end{aligned} \tag{3.56}$$

$$\begin{aligned}\frac{\partial^2 \ln(p(x|\boldsymbol{\varphi}, c))}{\partial \varphi_B^2} =& \frac{2}{\sigma_w^2}\eta s(c)\cos(\varphi_B - \varphi_A) - \frac{2}{\sigma_w^2}\eta\Re\left\{x\,\mathrm{e}^{-\,\mathrm{j}\,\varphi_B}\right\} \\ & \tanh\left(\frac{2}{\sigma_w^2}\left(\eta\Re\left\{x\,\mathrm{e}^{-\,\mathrm{j}\,\varphi_B}\right\} + s(c)\Re\left\{x\,\mathrm{e}^{-\,\mathrm{j}\,\varphi_A}\right\}\right)\right) + \\ & \left[\frac{\frac{2}{\sigma_w^2}\eta\Im\left\{x\,\mathrm{e}^{-\,\mathrm{j}\,\varphi_B}\right\}}{\cosh\left(\frac{2}{\sigma_w^2}\left(\eta\Re\left\{x\,\mathrm{e}^{-\,\mathrm{j}\,\varphi_B}\right\} + s(c)\Re\left\{x\,\mathrm{e}^{-\,\mathrm{j}\,\varphi_A}\right\}\right)\right)}\right]^2\end{aligned} \tag{3.57}$$

$$
\begin{aligned}
\frac{\partial^2 \ln(p(x|\boldsymbol{\varphi}, c))}{\partial\varphi_A \partial\varphi_B} =& \frac{-2}{\sigma_w^2}\eta s(c)\cos(\varphi_A - \varphi_B) + \frac{2}{\sigma_w^2}\Im\left\{x\,\mathrm{e}^{-\mathrm{j}\varphi_A}\right\} \\
& \frac{\frac{2}{\sigma_w^2}\eta s(c)\Im\left\{x\,\mathrm{e}^{-\mathrm{j}\varphi_B}\right\}}{\cosh\left(\frac{2}{\sigma_w^2}\left(\Re\left\{x\,\mathrm{e}^{-\mathrm{j}\varphi_A}\right\} + \eta s(c)\Re\left\{x\,\mathrm{e}^{-\mathrm{j}\varphi_B}\right\}\right)\right)^2}.
\end{aligned}
\tag{3.58}
$$

The final evaluation of the expectation is done using a Monte Carlo simulation in MATLAB, where we replace the expectation operator by the sample mean. To get the correct distribution we implement a simulation of the real system.

First we define constants, where N represents the observation length and NN the number of symbols used for averaging.

```
eta = 0.7;
PhiA = 0 * pi;
PhiB = 0.25 * pi;
SNR_dB_tab = 0:1:20;
N = 1e4;    %frame length
NN = 100e3; %length of integration
hA = 1 * exp(PhiA * 1j);
hB = eta * exp(PhiB * 1j);
```

Next we generate random data for both source nodes, map them using BPSK mappers and form their combined constellation vector **u**.

```
bA = logical(randi([0, 1], NN, 1));
bB = logical(randi([0, 1], NN, 1));
c = xor(bA, bB);
s = 1-2*c;
sA = 2*bA-1;
sB = 2*bB-1;
u = hA * sA + hB * sB;
J = zeros(2,2);
```

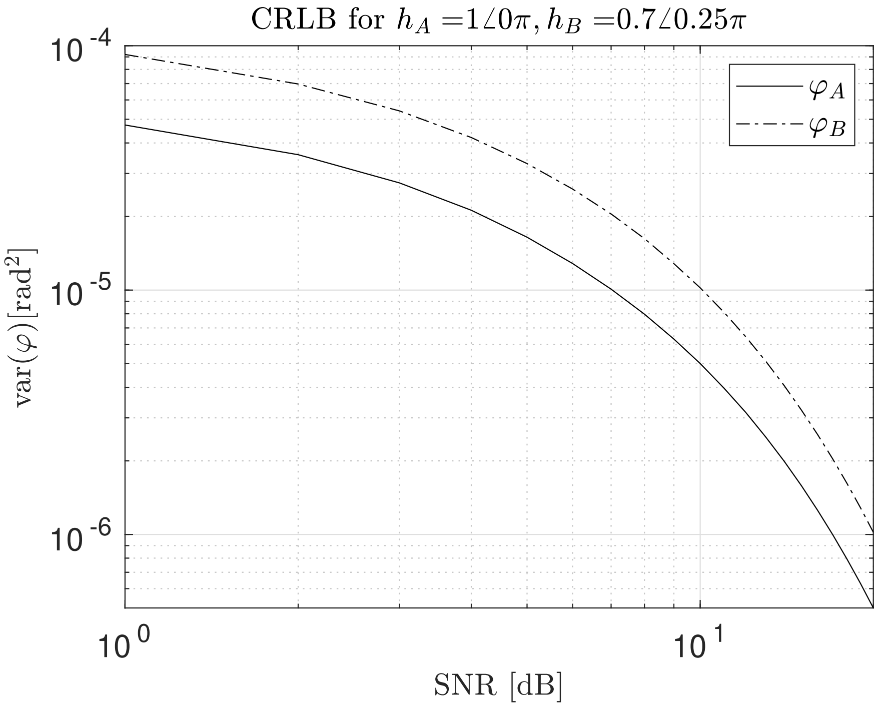


Figure 3.3: Monte Carlo numerical evaluation of the CRLB with $N = 10^4$.

For different SNR values, we evaluate the mean of the second order derivatives, form the Fisher information matrix $\mathbf{J}$ and compute its inverse. Its diagonal elements represent the bounds for an observation length of one. After division with the observation length N we get the final results.

```
for SNR_i = 1:length(SNR_dB_tab)
    SNR_dB = SNR_dB_tab(SNR_i);
    sigmaw2= 1/(10.^(SNR_dB/10));
    w = (randn(NN, 2) * [1; 1j]) * sqrt(sigmaw2/2);
    x = u + w;
    J(1,1) = -sum( ... )/NN;
    J(2,2) = -sum( ... )/NN;
    J(1,2) = -sum( ... )/NN;
    J(2,1) = J(1,2);
    invJ = inv(J);
    CBLRAsnr(SNR_i)= invJ(1,1)/N;
    CBLRBsnr(SNR_i)= invJ(2,2)/N;
end
```

The results for a particular parametrization $h_A = 1, h_B = 0.7 \exp(\mathrm{j}\,\frac{\pi}{4})$ are given in Figure 3.3. The difference between φ_A and φ_B directly corresponds to the inequality of the fading coefficient magnitudes.

Chapter 4

H-MAC Estimator Design and Performance Analysis

In this part we build on the theoretical conclusions from the previous chapter and propose complete estimator algorithms. We implement them using MATLAB and measure their performance using a H-MAC stage computer simulation.

4.1 Constant phase estimator

In this section we start with a less practical algorithm, which assumes a constant H-MAC channel parametrization over the whole frame. It serves mainly as a proof of concept, where it should theoretically be able to attain the CRLB derived in Section 3.4.5. Since it is not analytically tractable to obtain the maximum of 3.21 in closed form, we will use an iterative gradient search with additive updates. At i-th iteration the equation is given as

$$\hat{\boldsymbol{\varphi}}_{i+1} = \hat{\boldsymbol{\varphi}}_i + \mathbf{K}_1 \boldsymbol{\mu}_N, \tag{4.1}$$

where $\mathbf{K}_1$ is a properly chosen step-size diagonal matrix and

$$\boldsymbol{\mu}_N = \nabla_{\boldsymbol{\varphi}} \rho_N(\hat{\boldsymbol{\varphi}}_i, \mathbf{c}) = \sum_{n=1}^{N} \nabla_{\boldsymbol{\varphi}} \rho(\hat{\boldsymbol{\varphi}}_i, c_n) \tag{4.2}$$

is the gradient with symbol-wise components $\mu_{N,A} = \frac{\partial \rho_N}{\partial \varphi_A}, \mu_{N,B} = \frac{\partial \rho_N}{\partial \varphi_B}$. Using the partial derivatives derived in Section 3.4.5 we get

$$\begin{aligned}\mu_{N,A} = &\sum_{n=1}^{N} \frac{2}{\sigma_w^2}\eta s(c_n)\sin(\varphi_A - \varphi_B) + \frac{2}{\sigma_w^2}\Im\left\{x_n \,\mathrm{e}^{-\mathrm{j}\,\varphi_A}\right\} \\ &\tanh\left(\frac{2}{\sigma_w^2}\left(\Re\left\{x_n \,\mathrm{e}^{-\mathrm{j}\,\varphi_A}\right\} + s(c_n)\eta\Re\left\{x_n \,\mathrm{e}^{-\mathrm{j}\,\varphi_B}\right\}\right)\right),\end{aligned} \tag{4.3}$$

$$\begin{aligned}\mu_{N,B} = &\sum_{n=1}^{N} \frac{2}{\sigma_w^2}\eta s(c_n)\sin(\varphi_B - \varphi_A) + \frac{2}{\sigma_w^2}\eta\Im\left\{x_n \,\mathrm{e}^{-\mathrm{j}\,\varphi_B}\right\} \\ &\tanh\left(\frac{2}{\sigma_w^2}\left(\eta\Re\left\{x_n \,\mathrm{e}^{-\mathrm{j}\,\varphi_B}\right\} + s(c_n)\Re\left\{x_n \,\mathrm{e}^{-\mathrm{j}\,\varphi_A}\right\}\right)\right).\end{aligned} \tag{4.4}$$

For the gradient search to converge, the step-size is crucial. When chosen too small, the convergence may be very slow, when large the algorithm may not converge at all. The overall loop gain at the stable node is given as

$$\mathbf{K} = \mathbf{K}_1\mathbf{K}_0, \tag{4.5}$$

where $\mathbf{K}_0$ is a diagonal matrix containing the target function derivative magnitudes at the stable node, in our case at $\boldsymbol{\varphi} = \boldsymbol{\phi}$. Based on the hessian matrix from Equation 3.33, fixed by the multiplicative constant $\frac{N}{\sigma_w^2}$ we get

$$\mathbf{K}_0 = \begin{bmatrix} K_{0,A} & 0 \\ 0 & K_{0,B} \end{bmatrix} = \begin{bmatrix} \left|\left|\frac{\partial \mu_A}{\partial \varphi_A}\right|\right|_{\varphi_A = \phi_A} & 0 \\ 0 & \left|\left|\frac{\partial \mu_B}{\partial \varphi_B}\right|\right|_{\varphi_B = \phi_B} \end{bmatrix} = \begin{bmatrix} \frac{2N}{\sigma_w^2} & 0 \\ 0 & \frac{2N}{\sigma_w^2}\eta^2 \end{bmatrix}. \tag{4.6}$$

We see the relative scaling caused by the unequal fading coefficient magnitudes and a strong influence of the SNR and observation length. To chancel out these effects we set

$$\mathbf{K}_1 = K\mathbf{K}_0^{-1}, \tag{4.7}$$

where is K the step-size constant.

The final estimation procedure is given in Algorithm 1. The constants K and $\epsilon \ll 1$, the convergence threshold, need to be set properly. The algorithm runs the gradient search and then compares the metric at its convergent point with its values at other maxima and saddle points using the knowledge of the metric function structure derived in Section 3.4.3. The saddle points are included since in their neighborhood, the gradient may be very flat and thus the threshold condition may get triggered erroneously.

Algorithm 1 Constant phase estimator

```
 1: procedure ESTIMATOR1($\mathbf{x}, \mathbf{c}, \mathbf{K}_0$)
 2:     $\mathbf{K}_1 \leftarrow K\mathbf{K}_0^{-1}$                    ▷ $K$ is a given step-size constant
 3:     $i \leftarrow 0$
 4:     $\hat{\boldsymbol{\varphi}}_i \leftarrow [0,0]^T$
 5:     repeat
 6:         $\boldsymbol{\mu}_N \leftarrow \boldsymbol{\mu}_N(\hat{\boldsymbol{\varphi}}_i, \mathbf{x}, \mathbf{c})$
 7:         $\hat{\boldsymbol{\varphi}}_{i+1} \leftarrow \hat{\boldsymbol{\varphi}}_i + \mathbf{K}_1\boldsymbol{\mu}_N$
 8:         $i \leftarrow i+1$
 9:     until $\boldsymbol{\mu}_N^T\mathbf{K}_1\mathbf{K}_1\boldsymbol{\mu}_N < \epsilon$                    ▷ $\epsilon$ is a given threshold
10:     $A \leftarrow \left[\ \hat{\boldsymbol{\varphi}}_i, \quad \mathrm{flip}(\hat{\boldsymbol{\varphi}}_i), \quad \hat{\boldsymbol{\varphi}}_i + [0,\pi]^T, \quad \mathrm{flip}(\hat{\boldsymbol{\varphi}}_i) + [0,\pi]^T\ \right]^T$
11:     $M \leftarrow \rho_N(A, \mathbf{c})$
12:     $j \leftarrow \mathrm{getMaxIndex}(M)$
13:     if $j \neq 1$ then
14:         $\hat{\boldsymbol{\varphi}}_i \leftarrow A[j]$
15:         go to 5
16:     return $\hat{\boldsymbol{\varphi}}_i$
```

Implementation

In this section we provide the details of our MATLAB implementation of Algorithm 1.

First we set the initial estimates to 0 and compute the diagonal elements of $\mathbf{K}_1$ according to Equation 4.7.

```
phiA = 0;
phiB = 0;
K1_1 = sigmaw2/2*K/N;
K1_2 = sigmaw2/2*K/N/eta^2;
```

Then we start the iterative estimation. In each iteration, we compute the gradient for the actual phase estimate and apply an additive update.

```
while true
    muA= sum(2/sigmaw2*eta*s*sin(phiA-phiB)+ ...
        2/sigmaw2*imag(x*exp(-1j*phiA)).* ...
        tanh(2/sigmaw2*(real(x*exp(-1j*phiA))+ ...
        eta*s.*real(x*exp(-1j*phiB)))));
    muB= sum(2/sigmaw2*eta*s*sin(phiB-phiA)+ ...
        2/sigmaw2*eta*imag(x*exp(-1j*phiB)).* ...
        tanh(2/sigmaw2*(eta*real(x*exp(-1j*phiB))+ ...
        s.*real(x*exp(-1j*phiA)))));
    phiA = phiA + K1_1*muA;
    phiB = phiB + K1_2*muB;
```

If the convergence condition is met, we compose vector **A** containing the actual estimate and other stationary points derived from it. In **M** we store the corresponding metric values. Then we check whether the maximal value of **M** corresponds to the actual estimate. If not, we use the point corresponding to the maximum metric value as the new estimate and continue with the loop. In the other case we finish.

```
    if abs(K1_1*muA)^2 + abs(K1_2*muB)^2 < epsilon
        A = [phiA,phiB; phiB,phiA; phiA,phiB+pi; phiB,phiA+pi];
        M = zeros(1,4);
        for i = 1:4
            M(i) = sum(-2/sigmaw2*eta*s*cos(A(i,1)-A(i,2))+...
                   log(cosh(2/sigmaw2* ...
                   (real(x*exp(-1j*A(i,1)))+eta*s.* ...
                   real(x*exp(-1j*A(i,2)))))));
        end
        [~, j] = max(M);
        if j == 1
            [phiA, phiB] = equivalent_shift(phiA, phiB);
            break
        else
            phiA = A(j,1);
            phiB = A(j,2);
        end
    end
end
```

In order to get a unambiguous output in terms of Section 3.2.1, we use the function *equivalent_shift*, which shifts the result such that it lies in $\mathbb{P}$. The implementation is given as follows.

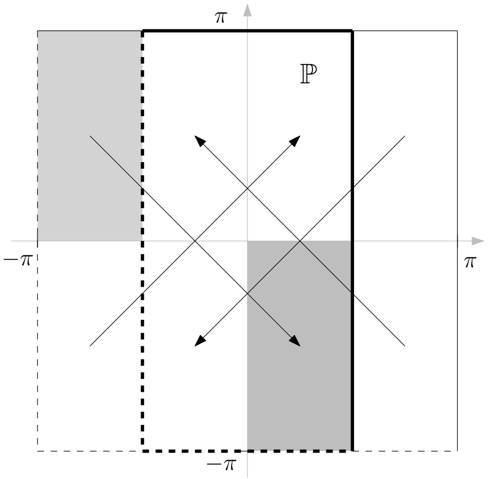


Figure 4.1: Illustration of the procedure of the *equivalent_shift* function. Full lines mark closed boundaries and dashed lines open boundaries.

First we apply an equivalent shift by $[(2k_1+1)\pi, (2k_2+1)\pi]^T, k_1, k_2 \in \mathbb{Z}$, such that both components lie in the interval $(-\pi; \pi]$ (outer rectangle in Figure 4.1).

```
function[phiA, phiB]= equivalent_shift(phiA, phiB)
    phiA = phiA - ceil(phiA/(2*pi))*2*pi + pi;
    phiB = phiB - ceil(phiB/(2*pi))*2*pi + pi;
```

Then again, using an equivalent shift, with $k_1, k_2 \in \{-1, 0\}$, we place the vector into $\mathbb{P}$ as shown in Figure 4.1.

```
    if phiA <= -pi/2
        phiA = phiA + pi;
        if phiB > 0
            phiB = phiB - pi;
        else
            phiB = phiB + pi;
        end
    end
    if phiA > pi/2
        phiA = phiA - pi;
        if phiB > 0
            phiB = phiB - pi;
        else
            phiB = phiB + pi;
        end
    end
end
```

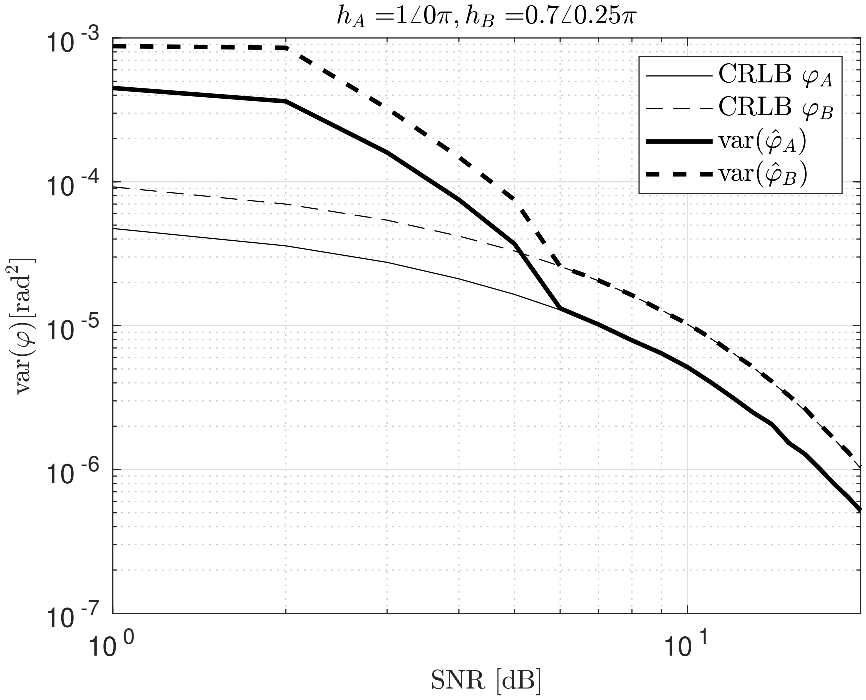


Figure 4.2: Comparison of a constant phase estimate with the CRLB. (Algorithm 1 with $K = 1, \epsilon = 0.01$)

Numerical results

In Figures 4.2 and 4.3 we present the results of a simulation of Algorithm 1, in terms of the achieved variance over multiple realizations. We plot the variance in dependence of the used SNR for both components of $\hat{\boldsymbol{\varphi}}$. For comparison we included the theoretical lower bound derived in Section 3.4.5. As expected, the variance depends highly on the value of the convergence threshold ϵ and can be brought to the theoretical limit, provided the supplied hierarchical target data are reliable.

In Figure 4.4 we evaluated how the variance of the estimates changes when we loosen the assumption of reliable target data estimates. For this evaluation we chose a fixed SNR equal to 20 dB. The unreliability of the provided hierarchical data was introduced by the insertion of a binary symmetric channel with the parameter p. It gives the probability of each bit being independently flipped.

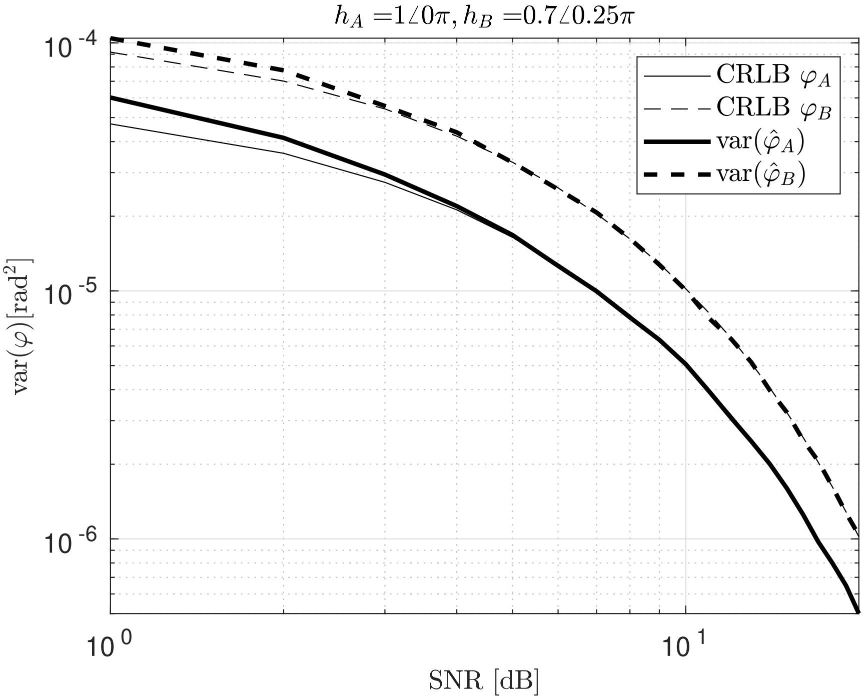


Figure 4.3: Comparison of a constant phase estimate with the CRLB. (Algorithm 1 with $K = 1, \epsilon = 0.0025$)

4.2 Variable phase estimator

Now we will loosen our strict assumption of a constant channel phase over the whole frame. Rather we assume the phase to be known at the start of the frame and the goal is to track the phase drift over the rest of the frame. For this purpose we propose a modification of Algorithm 1.

Algorithm 2 Variable phase, true data estimator

```
procedure ESTIMATOR2($\mathbf{x}, \mathbf{c}, \mathbf{K}_0, \boldsymbol{\varphi}$)
    $\mathbf{K}_1 \leftarrow K\mathbf{K}_0^{-1}$                ▷ $K$ is a given step-size constant
    $w \leftarrow \mathrm{floor}(\frac{W}{2})$                ▷ $W$ is a given averaging window length
    $s \leftarrow 1$
    $\hat{\boldsymbol{\varphi}}(0) \leftarrow \boldsymbol{\varphi}$
    while $s \leq \mathrm{len}(\mathbf{x}) + w$ do
        $n_l \leftarrow \max\{1, s - W\}$
        $n_h \leftarrow \min\{\mathrm{len}(x), s\}$
        $\boldsymbol{\mu}_s \leftarrow \frac{1}{n_h - n_l} \sum_{n=n_l}^{n_h} \boldsymbol{\mu}(\hat{\boldsymbol{\varphi}}(s-1), x_n, c_n)$
        $\hat{\boldsymbol{\varphi}}(s) \leftarrow \hat{\boldsymbol{\varphi}}(s-1) + \mathbf{K}_1\boldsymbol{\mu}_s$
        $s \leftarrow s + 1$
    return $[\hat{\boldsymbol{\varphi}}(w), \hat{\boldsymbol{\varphi}}(w+1), \ldots, \hat{\boldsymbol{\varphi}}(\mathrm{len}(\mathbf{x}) + w)]^T$
```

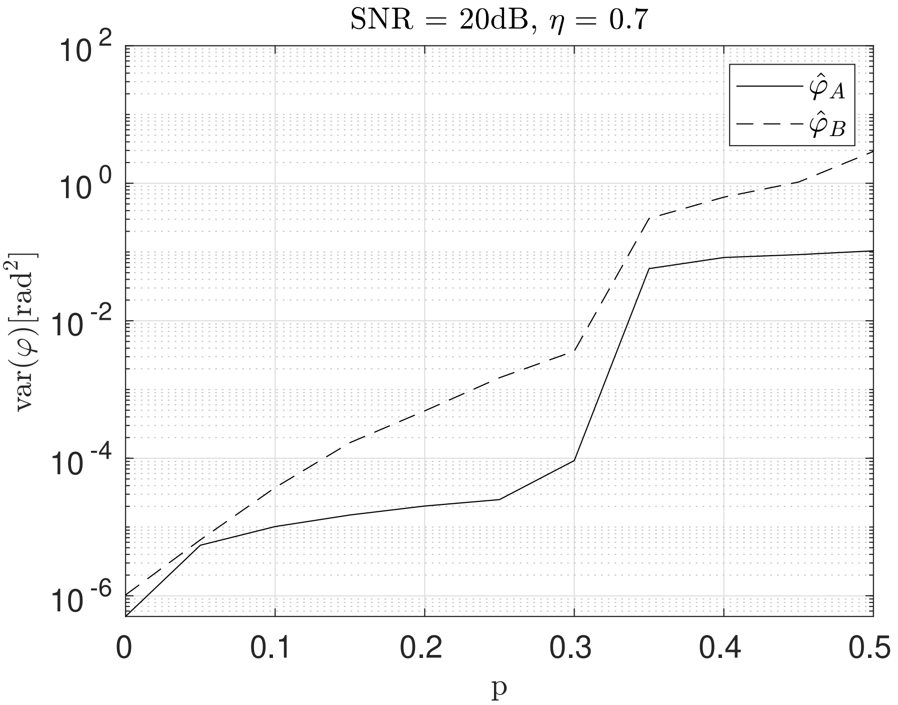


Figure 4.4: Dependence of the MSE on the data aid quality. Probability of erroneous data given by p. (Algorithm 1 with $K = 1, \epsilon = 0.0025$)

The proposed Algorithm 2 is parameterized by two constants. The step-size K serves the same purpose as for Algorithm 1. The window length W sets the dynamical properties. We assume the phase to be quasi-constant over W consecutive symbols and compute the gradient only over this window. Qualitatively we can say, that for smaller values of W, the estimator is able to track a faster changing parameter but is more susceptible to noise.

Implementation

In this section we provide the details of our Matlab implementation of Algorithm 2.

We assume, that at the start of the frame, both φ_A, φ_B are known to be $[0, 0]^T$. First we set the initial estimates to 0 and compute the diagonal elements of $\mathbf{K}_1$ according to Equation 4.7.

```
phiA = zeros(1,N+w);
phiB = zeros(1,N+w);
K1_1 = sigmaw2/2*K/N;
K1_2 = sigmaw2/2*K/N/eta^2;
```

Then we iterate over the whole frame. For each symbol we first determine the lower and upper index and cut out a window of length W.

```
for n = 2:N+w
    n_l = n - W;
    if n_l < 1
        n_l = 1;
    end
    n_h = n;
    if n_h > N
        n_h = N;
    end
    s_ = 1-2*c_est(n_l:n_h);
    x_ = x(n_l:n_h);
```

Next we compute a window-wide gradient and obtain the new estimate by applying an additive update to the previous one.

```
    av_len = (n-1-n_l+1);
    first_term = sum(2/sigmaw2*eta*s_* ...
                 sin(phiA(n-1)-phiB(n-1)))/av_len;
    muA= first_term ...
        +sum(2/sigmaw2*imag(x_*exp(-1j*phiA(n-1)))...
        .*tanh(2/sigmaw2*(real(x_*exp(-1j*phiA(n-1)))...
        +eta*s_.*real(x_*exp(-1j*phiB(n-1))))))/av_len;
    muB= -first_term ...
        +sum(2/sigmaw2*eta*imag(x_*exp(-1j*phiB(n-1)))...
        .*tanh(2/sigmaw2*(eta*real(x_*exp(-1j*phiB(n-1)))...
        +s_.*real(x_*exp(-1j*phiA(n-1))))))/av_len;
    phiA(n) = phiA(n-1) + K1_1*muA;
    phiB(n) = phiB(n-1) + K1_2*muB;
end
```

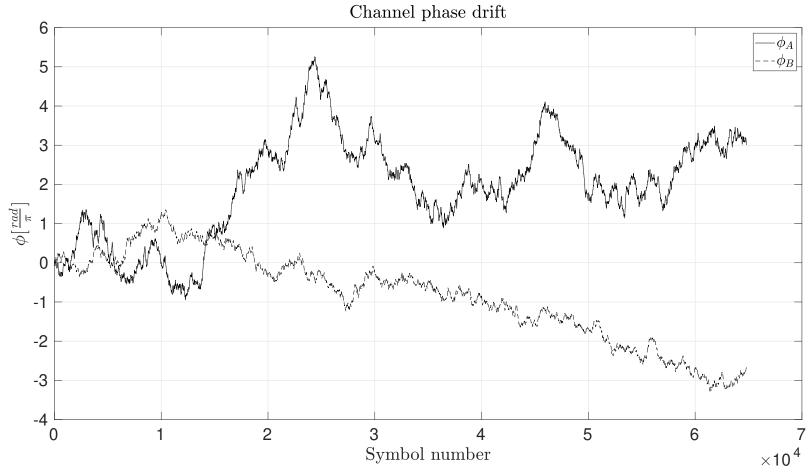


Figure 4.5: Example of simulated H-MAC channel phase drift.

Finally we remove the first w estimates to equalize the introduced delay.

```
for n = 1:w
    phiA(n) = [];
    phiB(n) = [];
end
```

Numerical results

To evaluate the characteristics of the proposed Algorithm 2, we run a simulation of the H-MAC channel with a randomly time varying phase. The phase drift is simulated as a random walk. One realization of the H-MAC channel phase course over a frame of length $N = 64800$ is depicted in Figure 4.5.

The estimation algorithm was run with the step-size and average window length experimentally set to $K = 0.4N, W = 20$. In Figure 4.6 we see a cutout of one realization of the phase course ϕ_A, ϕ_B and the estimator output $\hat{\varphi}_A, \hat{\varphi}_B$ at low SNR. After symbol number $n = 1.1 \cdot 10^4$ we see a transition between two lock points. From the theoretical part we know that both are equivalent in terms of Equation 3.5 where for this particular case $k_1 = -1, k_2 = 1$.

In Figure 4.8 we see again a cutout from a particular realization. In this case η was set to 0.3 which demonstrates trough the significantly uneven performance between both $\hat{\varphi}_A, \hat{\varphi}_B$.

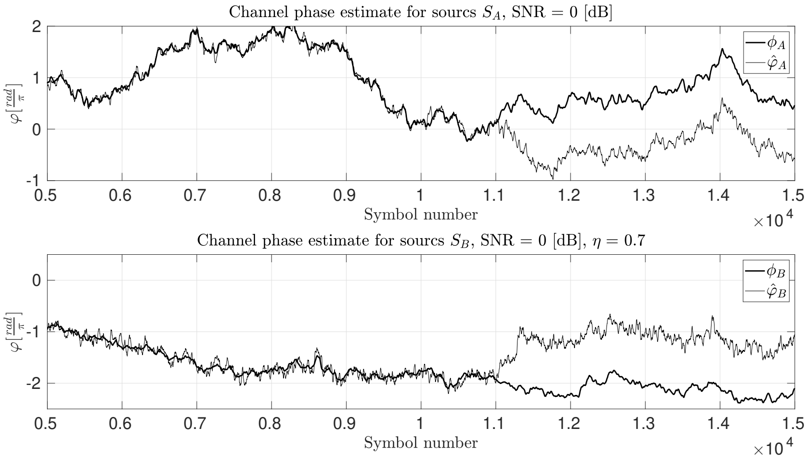


Figure 4.6: Example of perfect H-data aided estimator phase tracking with transition between equivalent stable lock points. (Algorithm 2 with $N = 64800$, $\frac{K}{N} = 0.4, W = 20$)

The overall performance was evaluated in terms of the mean square error of the phase estimates. During the evaluation, the phase invariance w.r.t. the hierarchical symbol was taken into account. The results are shown in Figure 4.8. The frame length was also set to $N = 64800$. In the figure we see the relationship of the estimate MSE on the SNR for two cases. In the first we used reliable data estimates, in the later we introduced in each bit an error with the probability p. We observe that the sensitivity to erroneous data aid is lower for low SNR where the noise is the main limiting factor.

4.3 Front-end demodulation loop

In this section we consider a very practical scenario, we do not require neither the data nor the channel phase to be know. We assume only the SNR and the individual channel coefficient magnitudes to be known, as well as the initial phase. This can be easily obtained from a short pilot based channel measurement phase which precedes the payload transmission. This phase seems unavoidable even from the point of detection and classification of individual H-MAC stage participants and their channel qualities.

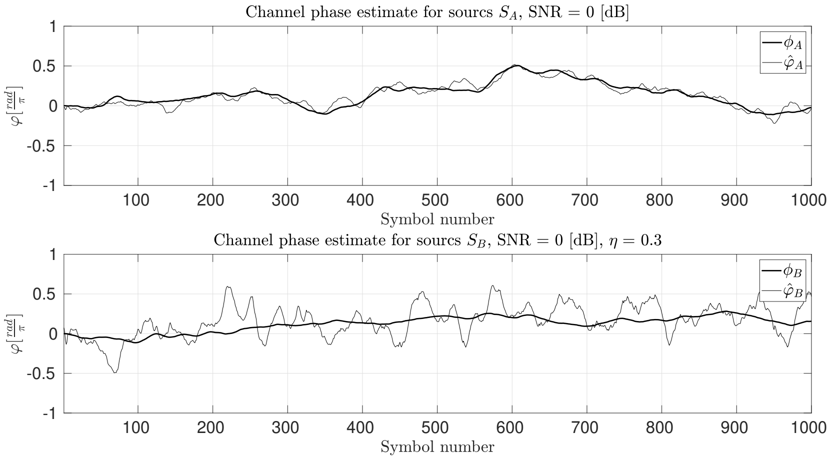


Figure 4.7: Example of perfect H-data aided estimator phase tracking with noticeable accuracy difference as a consequence of small η. (Algorithm 2 with $N = 64800$, $\frac{K}{N} = 0.4, W = 20$)

4.3.1 System without FEC

First we consider an uncoded system. In the terms of Section 2, this means that $\mathcal{C}$ is an identity and $\mu_c = \mu_b$. The system schema is show in Figure 4.9. Because the data is not encoded, there is no inter-symbol dependency and thus the iterative loop is executed for each symbol independently. The proposed procedure is given in Algorithm 3. We start with a zero data estimate and an initial phase estimate given from the pilot phase and iterate for each symbol the H-SODEM - H-CSE loop (loop 1) I times. The H-CSE is based on the procedure given in Algorithm 2.

Implementation

In this section we provide the details of our MATLAB implementation of Algorithm 3. First we present the implementation of the SODEM derived in Section 3.3.

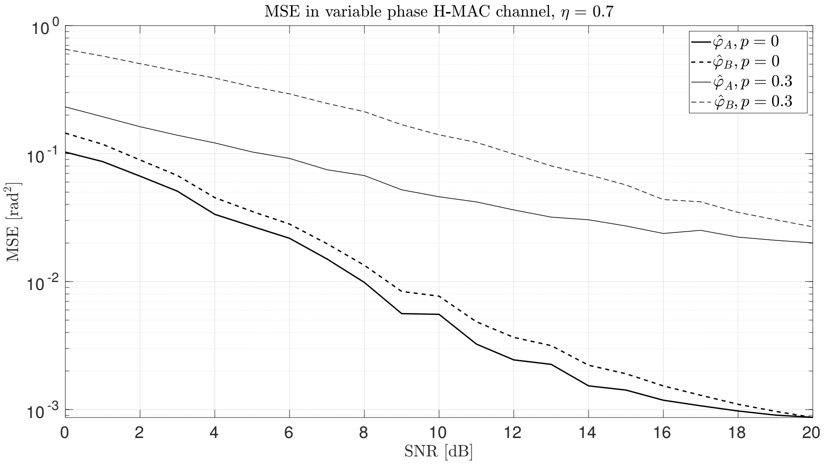


Figure 4.8: H-data aided estimator MSE. Probability of erroneous data given by p. (Algorithm 2 with $N = 64800$, $\frac{K}{N} = 0.4, W = 20$)

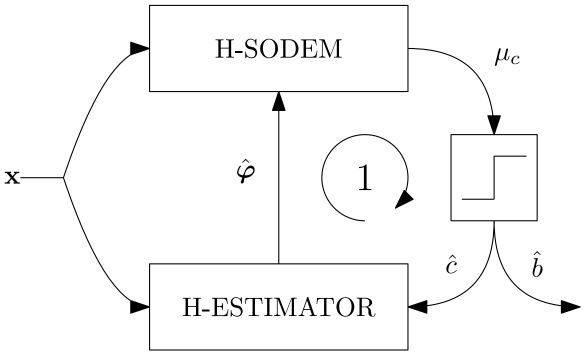


Figure 4.9: Schema of a node front-end processing with uncoded input.

The function is based on Equation 3.11. Based on the given channel parametrization we compute the combined constellation points for all four source symbol combinations. Then, the scaled likelihoods for $c_n = 1$ and $c_n = 0$ are given as a sum of the corresponding Gaussian densities. Finally the log-likelihood ratio is computed.

```
function LLR = sodem(x, sigmaw2, phiA, phiB, eta)
    u00 = -(exp(1j*phiA)+eta*exp(1j*phiB));
    u01 = -exp(1j*phiA)+eta*exp(1j*phiB);
    u10 = -u01;
    u11 = -u00;
    p0 = exp(-1/sigmaw2*abs(x-u00).^2) + ...
         exp(-1/sigmaw2*abs(x-u11).^2);
    p1 = exp(-1/sigmaw2*abs(x-u01).^2) + ...
         exp(-1/sigmaw2*abs(x-u10).^2);
    LLR = log(p0./p1);
end
```

Algorithm 3 Variable phase uncoded signal receiver loop

```
procedure RECEIVER1($\mathbf{x}, \mathbf{K}_0, \boldsymbol{\varphi}$)
    $\mathbf{K}_1 \leftarrow K\mathbf{K}_0^{-1}$                          ▷ $K$ is a given step-size constant
    $w \leftarrow \mathrm{floor}(\frac{W}{2})$                 ▷ $W$ is a given averaging window length
    $s \leftarrow 1$
    $\hat{\boldsymbol{\varphi}}(0) \leftarrow \boldsymbol{\varphi}$
    $\hat{\mathbf{c}} \leftarrow \mathbf{0}$
    while $s \leq \mathrm{len}(\mathbf{x}) + w$ do
        $n_l \leftarrow \max\{1, s - W\}$
        $n_h \leftarrow \min\{\mathrm{len}(x), s\}$
        $i \leftarrow 1$
        $\hat{\boldsymbol{\varphi}}_i \leftarrow \hat{\boldsymbol{\varphi}}(s-1)$
        while $i \leq I$ do      ▷ $I$ is a given number of iterations pro symbol
            $\hat{c}_{n_h} = \mathrm{decide}(\mathrm{SODEM}(x_{n_h}, \boldsymbol{\varphi}_i))$
            $\boldsymbol{\mu}_s \leftarrow \frac{1}{n_h - n_l} \sum_{n=n_l}^{n_h} \boldsymbol{\mu}(\hat{\boldsymbol{\varphi}}(s-1), x_n, \hat{c}_{n_h})$
            $\hat{\boldsymbol{\varphi}}_{i+1} \leftarrow \hat{\boldsymbol{\varphi}}_i + \mathbf{K}_1 \boldsymbol{\mu}_s$
            $i \leftarrow i + 1$
        $\hat{\boldsymbol{\varphi}}(s) \leftarrow \hat{\boldsymbol{\varphi}}_I$
        $s \leftarrow s + 1$
    return $\hat{\mathbf{c}}$
```

The implementation of loop 1 from 4.9 is given as follows.

The initialization stays the same.

```
phiA = zeros(1,N+w);
phiB = zeros(1,N+w);
K1_1 = sigmaw2/2*K/N;
K1_2 = sigmaw2/2*K/N/eta^2;
```

The loop is also very similar as for the pure variable phase estimator. The main difference is made by replacing the data aid by the output of the SODEM. To obtain a hard decision from the LLR SODEM output, we simply compare it with zero. For each symbol, we run the inner loop (loop 1 in 4.9) *I* times.

```
for n = 2:N+w
    for i = 1:I
        n_l = n - W;
        if n_l < 1
            n_l = 1;
        end
        n_h = n;
        if n_h > N
            n_h = N;
        else
            c_est(n) = ...
            (sodem(x(n), sigmaw2, phiA(n), phiB(n), eta) < 0);
        end
        s_ = 1-2*c_est(n_l:n_h);
        x_ = x(n_l:n_h);
        av_len = (n-1-n_l+1);
        first_term = sum(2/sigmaw2*eta*s_* ...
        sin(phiA(n-1)-phiB(n-1)))/av_len;
        muA= first_term ...
        +sum(2/sigmaw2*imag(x_*exp(-1j*phiA(n-1)))...
        .*tanh(2/sigmaw2*(real(x_*exp(-1j*phiA(n-1)))...
        +eta*s_.*real(x_*exp(-1j*phiB(n-1))))))/av_len;
        muB= -first_term ...
        +sum(2/sigmaw2*eta*imag(x_*exp(-1j*phiB(n-1)))...
        .*tanh(2/sigmaw2*(eta*real(x_*exp(-1j*phiB(n-1)))...
        +s_.*real(x_*exp(-1j*phiA(n-1))))))/av_len;
        phiA(n) = phiA(n-1) + K1_1*muA;
        phiB(n) = phiB(n-1) + K1_2*muB;
    end
end
for n = 1:w
    phiA(n) = [];
    phiB(n) = [];
end
c_est = (sodem(x, sigmaw2, phiA’, phiB’, eta) < 0);
```

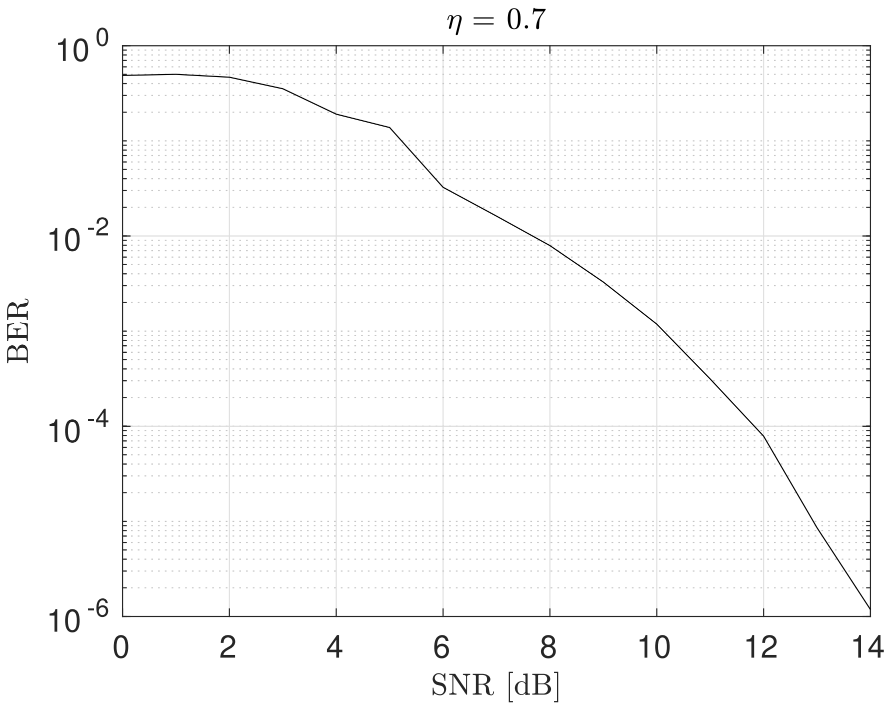


Figure 4.10: Bit error rate at different SNR for uncoded system.

Numerical results

In Figure 4.10 we present the result of a simulation based on the schema in Figure 4.9. The same H-MAC channel model with the drifting phase as in Section 4.2 was used. Even without the knowledge of the hierarchical data, nor the channel phase parametrization, the SODEM-CSE loop converges. The graph shows the measured *bit error rates* (BER) for different values of SNR. We see that even without an forward error correcting code, the bit error rate drops under 10^{-6} for SNR > 14dB.

4.3.2 System with FEC

Now we extend the system presented in Section 4.3.1 with a FEC. We use an LDPC code from the DVB-S2 standard with rate $R = \frac{1}{2}$ and $N = 64800$. It is a linear code and thus by employing it, we obtain an isomorphic layered NCM. The system schema is shown in Figure 4.11. The first part comprised from the H-SODEM and H-CSE, together with loop 1, remained the same as for the uncoded version. Since the first loop runs symbol-wise, and thus does not exploit the inner signal structure, we add a second loop, which iterates

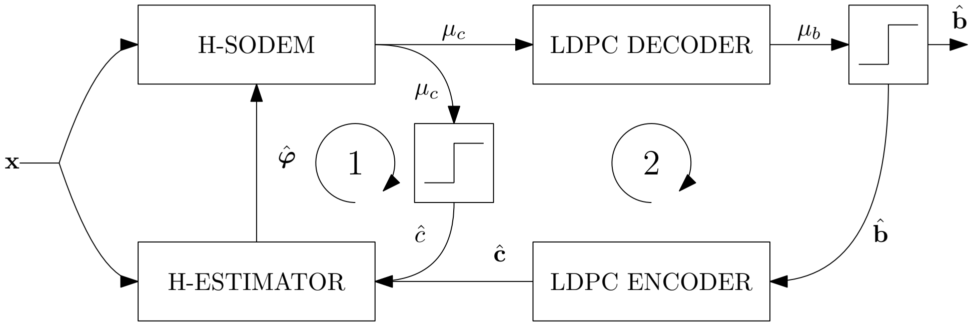


Figure 4.11: Schema of a node processing with encoded input.

the LDPC decoder and H-CSE on the whole frame vector.

The overall procedure is a combination of those introduced previously. First Algorithm 3 is run and then $\hat{\mathbf{c}}_1$ is fed trough the LDPC DEC-ENC pair and used as the input to Algorithm 2. The newly obtained phase estimate is used by the H-SODEM, providing an updated code-word estimate $\hat{\mathbf{c}}_2$. At this point the cycle repeats. The loop is iterated until the LDPC decoder is able to decode error-free or a predefined number of iterations is reached.

Implementation

In this section we provide the details of our Matlab implementation. The part implementing loop 1 from Figure 4.11 is the same as showed above and will not be repeated here. For the LDPC encoder and decoder in loop 2 we use the Matlab build in implementation.

First we instantiate the LDPC encoder and decoder using a parity matrix used in the DVB-S2 system.

```
hEnc = comm.LDPCEncoder('ParityCheckMatrix', dvbs2ldpc(rate));
hDec = comm.LDPCDecoder('ParityCheckMatrix', ...
       dvbs2ldpc(rate), 'FinalParityChecksOutputPort',true);
```

The implementation of loop 2 is as follows. First we get the estimate LLR from the SODEM and try to decode it. When the decoding succedes and all parity checks are met, we break the loop and return our estimate $\hat{\mathbf{b}}$. If not, we pass the hard decision (made by the LDPC encoder) to the encoder to get a new estimate $\hat{\mathbf{c}}$. Then we run the H-CSE and obtain an updated version of the channel phase estimate.

```
for i = 1:I
    c_est = sodem(x, sigmaw2, phiA', phiB', eta);
    [b_est, parity] = step(hDec, c_est);
    if sum(parity) == 0
        break
    end
    c_est = step(hEnc, b_est);
    for n = 2:N+w
        n_l = n - W;
        if n_l < 1
            n_l = 1;
        end
        n_h = n;
        if n_h > N
            n_h = N;
        else
            c_est(n) = ...
            (sodem(x(n), sigmaw2, phiA(n), phiB(n), eta) < 0);
        end
        s_ = 1-2*c_est(n_l:n_h);
        x_ = x(n_l:n_h);
        av_len = (n-1-n_l+1);
        first_term = sum(2/sigmaw2*eta*s_* ...
        sin(phiA(n-1)-phiB(n-1)))/av_len;
        muA= first_term ...
        +sum(2/sigmaw2*imag(x_*exp(-1j*phiA(n-1)))...
        .*tanh(2/sigmaw2*(real(x_*exp(-1j*phiA(n-1)))...
        +eta*s_.*real(x_*exp(-1j*phiB(n-1))))))/av_len;
        muB= -first_term ...
        +sum(2/sigmaw2*eta*imag(x_*exp(-1j*phiB(n-1)))...
        .*tanh(2/sigmaw2*(eta*real(x_*exp(-1j*phiB(n-1)))...
        +s_.*real(x_*exp(-1j*phiA(n-1))))))/av_len;
        phiA(n) = phiA(n-1) + K1_1*muA;
        phiB(n) = phiB(n-1) + K1_2*muB;
    end
    for n = 1:w
        phiA(n) = [];
        phiB(n) = [];
    end
end
```

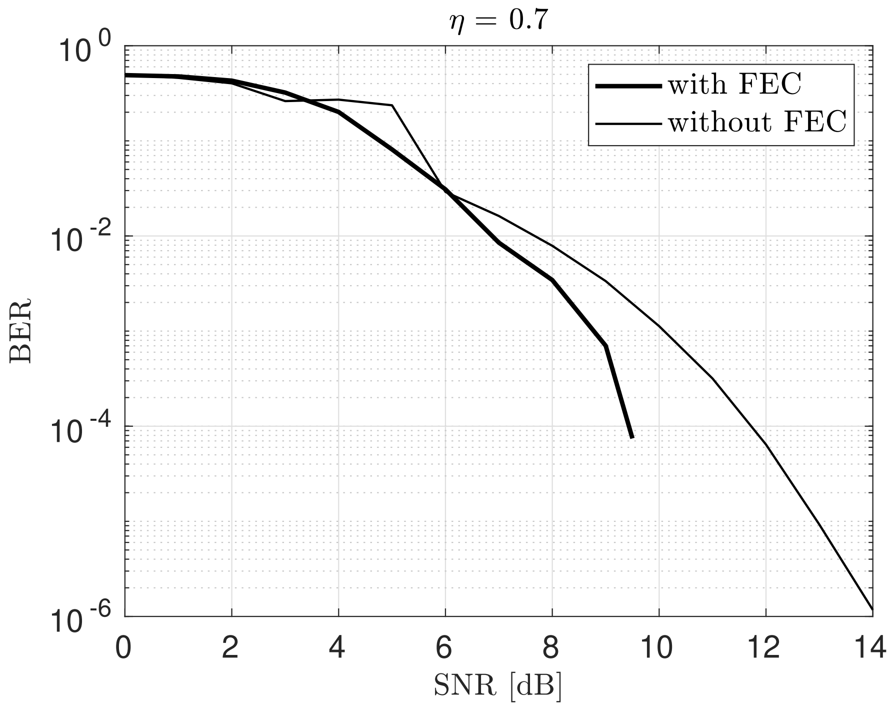


Figure 4.12: Bit error rate at different SNR for both cases, with and without FEC.

Numerical results

The simulation of the system, based on the schema in Figure 4.11 was run under the same conditions as for the uncoded case. The performance was also measured using the bit error rate. The results are presented in Figure 4.12. For comparison, we repeated the curve for the case without FEC. For the low SNR region, the performance is nearly the same. For higher SNR hovewer, we see the performance gain of the FEC. For LDPC codes it is typical, that the BER falls dramatically in a small SNR range.

Chapter 5

Conclusions

We introduced the fundamental principles of WPNC and explained the basic concepts and definitions. In particular, we addressed the overall WPNC radio network structure and the global solvability condition as well as the processing chain of individual nodes. Based on the gained understanding, we defined a particular scenario, assumed throughout the rest of the work. Namely a two-source H-MAC stage with BPSK modulation and XOR HNC map. We showed the relation of the hierarchical symbol and the channel phase parametrization, particularly the link between hierarchical dispersion and channel phase ambiguity. Further, we performed the derivation of the H-SODEM and hierarchical data decision aided ML phase estimator considering our specific scenario. The main theoretical contribution resulted from the analysis of the estimator metric in terms of the conditions of existence and positions of local maxima. Using numerical methods, we determined the theoretical limit of the ML estimator in terms of the CRLB.

Based on the theoretic results, we developed particular estimation algorithms. First we presented an estimator for the case of random, but constant channel phase over the observation frame. Using a simulation we showed, that it has the potential to attain the theoretical limit. Next we considered a channel model with time varying phase and proposed a modified algorithm, which is capable of tracking the phase drift. Based on a simulation, we obtained the tracking error for different SNR values and analyzed the performance drop caused by supplying unreliable hierarchical data estimates. In the end we implemented a complete receiver front-end, comprising from our H-CSE, a H-SODEM and a LDPC decoder and measured the resulting performance in terms of achieved BER.

Appendix A

Bibliography

[1] U. Spagnolini. *Statistical Signal Processing in Engineering*. John Wiley & Sons, 2018.

[2] J. Sykora. Hierarchical data decision aided 2-source bpsk h-mac channel phase estimator with feed-back gradient solver for wpnc networks. In *2018 14th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pages 89–96. IEEE, 2018.

[3] J. Sykora and A. Burr. *Wireless Physical Layer Network Coding*. Cambridge University Press, 2018.

Appendix B

Content of the CD-ROM

```
ROOT
├── matlab_scripts
│   ├── Constant_phase_estimator.m
│   ├── Constant_phase_estimator_unreliable_data_aid.m
│   ├── equivalent_shift.m
│   ├── H-CSE_CRLB.m
│   ├── sodem.m
│   ├── square_error.m
│   ├── Variable_phase_coded_system.m
│   ├── Variable_phase_estimator.m
│   └── Variable_phase_uncoded_system.m
├── results
│   └── workspace_*.mat
└── Thesis.pdf
```

ZADÁNÍ DIPLOMOVÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Hron** Jméno: **Petr** Osobní číslo: **434850**

Fakulta/ústav: **Fakulta elektrotechnická**

Zadávající katedra/ústav: **Katedra radioelektroniky**

Studijní program: **Otevřené elektronické systémy**

Studijní obor: **Komunikace a zpracování signálu**

II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

Estimace kanálu a síťově kódované modulace pro parametrický H-MAC kanál v WPNC rádiových sítích

Název diplomové práce anglicky:

Channel Estimation and Network Coded Modulation for Parametric H-MAC Channel in WPNC Radio Networks

Pokyny pro vypracování:

The student will get acquainted with the fundamentals of WPNC (Wireless Physical Layer Network Coding), design principles of NCM (Network Coded Modulation) and channel estimation algorithms for hierarchical MAC channel. The work goals are (1) a design and performance analysis of the H-MAC channel estimation algorithms for various forms of a priori data information form and/or contents, (2) an analysis of the impact of imperfect H-MAC channel estimation on the hierarchical detection of isomorphic layered NCM, and possibly also an attempt to design the NCM which would be more resistant to these errors. The scenario should include at least 2-source node topology with PSK modulation and some state-of-the-art outer code (e.g. LDPC). The performance analysis should include BER properties, channel estimation MSE performance and (if the feed-back or iterative solver is used) convergence properties.

Seznam doporučené literatury:

[1] Jan Sykora, Alister Burr: Wireless Physical Layer Network Coding, Cambridge University Press, 2018

Jméno a pracoviště vedoucí(ho) diplomové práce:

prof. Ing. Jan Sýkora, CSc., katedra radioelektroniky FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) diplomové práce:

Datum zadání diplomové práce: **28.01.2019** Termín odevzdání diplomové práce: **24.05.2019**

Platnost zadání diplomové práce: **20.09.2020**

prof. Ing. Jan Sýkora, CSc. podpis vedoucí(ho) práce	prof. Mgr. Petr Páta, Ph.D. podpis vedoucí(ho) ústavu/katedry	prof. Ing. Pavel Ripka, CSc. podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Diplomant bere na vědomí, že je povinen vypracovat diplomovou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v diplomové práci.

.

Datum převzetí zadání

Podpis studenta