Multiple–Symbol Differential Decision Fusion for Wireless Sensor Networks

by

Andre Lei

B.A.Sc, University of British Columbia, 2005

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

in

THE FACULTY OF GRADUATE STUDIES

(Electrical and Computer Engineering)

THE UNIVERSITY OF BRITISH COLUMBIA
(Vancouver)

April 2009

Abstract

We consider the problem of decision fusion in mobile wireless sensor networks where the channels between the sensors and the fusion center are time–variant. We assume that the sensors make independent local decisions on the M hypotheses under test and report these decisions to the fusion center using differential phase–shift keying (DPSK), so as to avoid the channel estimation overhead entailed by coherent decision fusion. For this setup we derive the optimal and three low–complexity, suboptimal fusion rules which do not require knowledge of the instantaneous fading gains. Since all these fusion rules exploit an observation window of at least two symbol intervals, we refer to them collectively as multiple–symbol differential (MSD) fusion rules. For binary hypothesis testing, we derive performance bounds for the optimal fusion rule and exact or approximate analytical expressions for the probabilities of false alarm and detection for all three suboptimal fusion rules. Simulation and analytical results confirm the excellent performance of the proposed MSD fusion rules and show that in fast fading channels significant performance gains can be achieve by increasing the observation window to more than two symbol intervals.

Contents

List of Figures

List of Abbreviations

AWGN	Additive white gaussian noise
BER	Bit error rate
BHT	Binary hypothesis testing
CSI	Channel state information
CV	Chair–Varshney
DD	Differential detection
DPSK	Differential phase–shift keying
FC	Fusion center
FSK	Frequency–shift keying
ILS	Ideal Local Sensor
MHT	Multiple hypothesis testing
ML	Maximum-likelihood
MSD	Multiple–symbol differential
MSDD	Multiple–symbol differential detection
MSDSD	Multiple–symbol sphere decoding
MSDHT	Multiple–symbol differential hypothesis testing
OOK	On–off keying
PBPO	Person by person optimization
pdf	Probability density function
PEP	Pairwise error probability
PSK	Phase–shift keying

TDMA	Time–division multiple access
ROC	Receiver operating characteristic
RTS	Repeated tree search
SD	Sphere decoding
SE	Schnoor–Euchner
SNR	Signal–to–noise ratio
STS	Single tree search
WSN	Wireless sensor network

Operators and Notations

$\lvert\cdot\rvert$	Absolute value of a complex number
$[\cdot]^*$	Complex conjugate
$[\cdot]^T$	Matrix or vector transposition
$[\cdot]^H$	Matrix or vector Hermitian transposition
$\Re\{\cdot\}$	Real part of a complex number
$\mathcal{E}\{\cdot\}$	Statistical expectation
$\delta(\cdot)$	Dirac delta function
$\bar{u}[\cdot]$	Unit step function
$[\boldsymbol{X}]_{i,j}$	Element of matrix $\boldsymbol{X}$ in row i and column j
$\det(\cdot)$	Determinant
$\boldsymbol{I}_X$	$X \times X$ identity matrix
$\otimes$	Kronecker product
$\mathrm{diag}\{\boldsymbol{X}_1, \ldots, \boldsymbol{X}_X\}$	Block diagonal matrix with matrices $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_X$ on its main diagonal
$P(\cdot)$	Probabilities
$p(\cdot)$	Probability density function

Acknowledgments

Foremost, I would like to thank my supervisor, Dr. Robert Schober, for the constant guidance and support that he has provided. Without his helpful suggestions, advice and constant encouragement, this work would not have been possible.

I would also like to thank my friends and colleagues for all the discussions, advice and laughter that we have shared throughout the years. Not only have you all helped me keep my perspective, but also my sanity during my ups and downs.

Finally, I would like to thank my family who have always encouraged me and given their unconditional support throughout my studies at the University of British Columbia.

ANDRE LEI

Chapter 1

Introduction

1.1 Wireless Sensor Networks

Wireless sensor networks (WSNs) have been a topic of much interest because of their miliary and civilian applications which include target tracking, environmental monitoring, control, and spectrum sensing [1–5]. Recently advances in the development of low-cost microsensors and an increased interest in cognitive radio have renewed interest in WSNs and the decision fusion problem [6, 7]. In civilian applications, low-cost microsensors can be deployed for environmental sampling, surveillance, and habitat monitoring, allowing WSNs to be used to solve various problems in different fields. For example, it has been shown that licensed spectrum is frequently underused [8]. Spectrum sensing can be employed to identify unused spectrum and thereby allow cognitive radios to operate when licensed users are idle, enabling more efficient use of the limited spectrum [4, 5, 9, 10]. Both spectrum sensing and cognitive radios depend on decision fusion, which provides a fault-tolerant and robust method of detection while also increasing detection performance by using observations from multiple sensors to arrive at a more reliable decision than can be achieved by individual sensors.

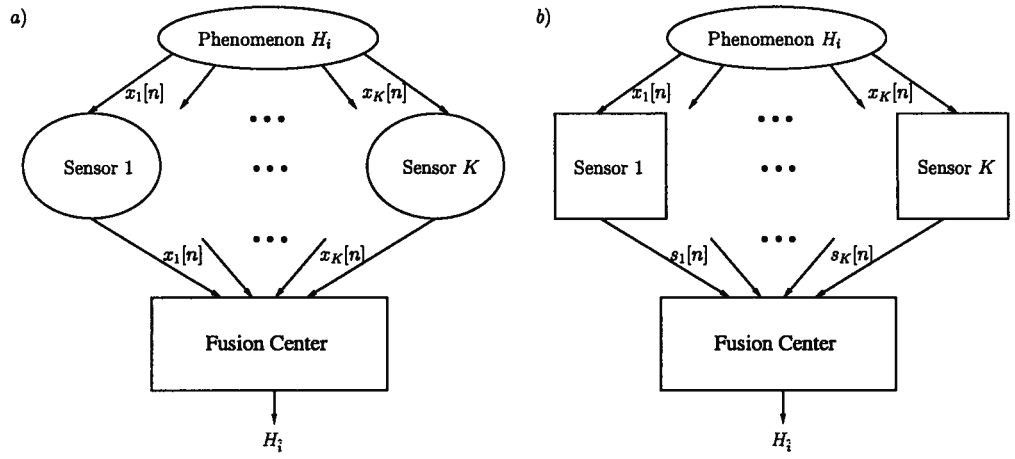

Figure 1.1: Wireless sensor network with K sensors: (a) Centralized scheme (b) Decentralized scheme.

WSNs can be centralized or decentralized. Fig. 1.1a shows a centralized sensor network containing K sensors and Fig. 1.1b shows a decentralized sensor network. In centralized detection, each sensor collects the information of interest denoted by $x_k[n]$ and transmits its raw observation to the fusion center (FC) for processing. Fusion within a centralized scheme is solved by classical hypothesis testing, which makes a global decision on the current phenomenon being observed, i.e., the hypothesis under test H_i, $i \in \mathcal{M}$, $\mathcal{M} \triangleq \{0, 1, \ldots, M-1\}$ [11]. Classical papers by Tenney, Sandell [12] and Tsitsiklis [13] introduced the decentralized detection problem, where sensors make independent decisions based on signal processing of their individual observations. The decision, $s_k[n]$, is then transmitted to the FC to arrive at a final estimate of $H_{\hat{i}}$, $i \in \mathcal{M}$. Since the processing performed by both the sensors and the FC is threshold based [13], optimizing the system requires jointly optimizing the thresholds at all the sensors and the FC simultaneously, and is thus a complex problem. To reduce the complexity, a person by person optimization (PBPO) method is frequently used [11], which satisfies the necessary condition for optimality but does not guarantee a global optimal solution. PBPO attempts to optimize the processing at each sensor independently by fixing the processing rules at all

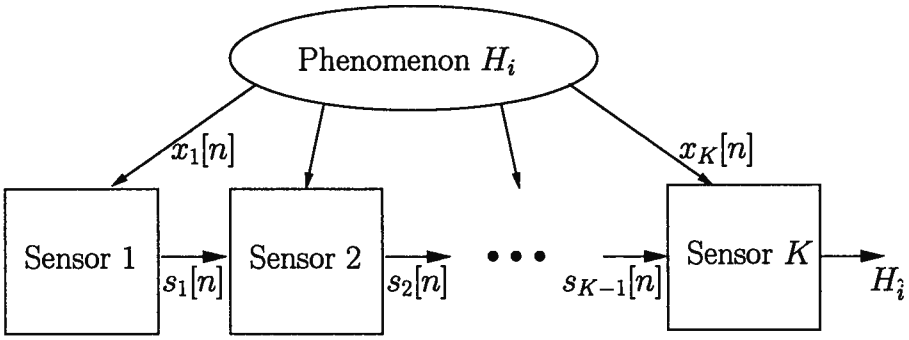

Figure 1.2: Serial topology sensor network.

other sensors and the FC while one sensor is being optimized. This process is repeated for the remaining $K-1$ sensors and the FC.

Although centralized detection performs best, each sensor is burdened with transmitting raw data to the FC, resulting in large power consumption and large communication bandwidth requirement. Therefore, in a wireless situation where sensors are battery operated and communication resources are limited, the decentralized scheme is of particular interest.

In general there are three common WSN topologies, namely parallel, serial and tree [1]. The parallel configuration in Fig. 1.1b is the most common type of WSN considered in literature. K sensors receive a noisy observation of the phenomenon H_i denoted by $x_k[n]$ and, after processing the information, transmit their local decisions, $s_k[n]$, to the FC. The FC then makes a global decision based on the decision of K sensors and not the observations of each individual sensor, $x_k[n]$.

Fig. 1.2 illustrates a general WSN in serial configuration. The 1st sensor within the WSN processes only its own observation whereas the remaining $K-1$ sensors process both their own observations and the decisions made by the previous sensors. Thus, the kth sensor will fuse its observations with the decision from the $(k-1)$th sensor to generate its own decision. The aggregated data, or the final decision, is presented by the Kth sensor of the WSN.

A WSN in a tree topology is shown in Fig. 1.3. In its simplest form, $\lceil \frac{K}{2} \rceil$

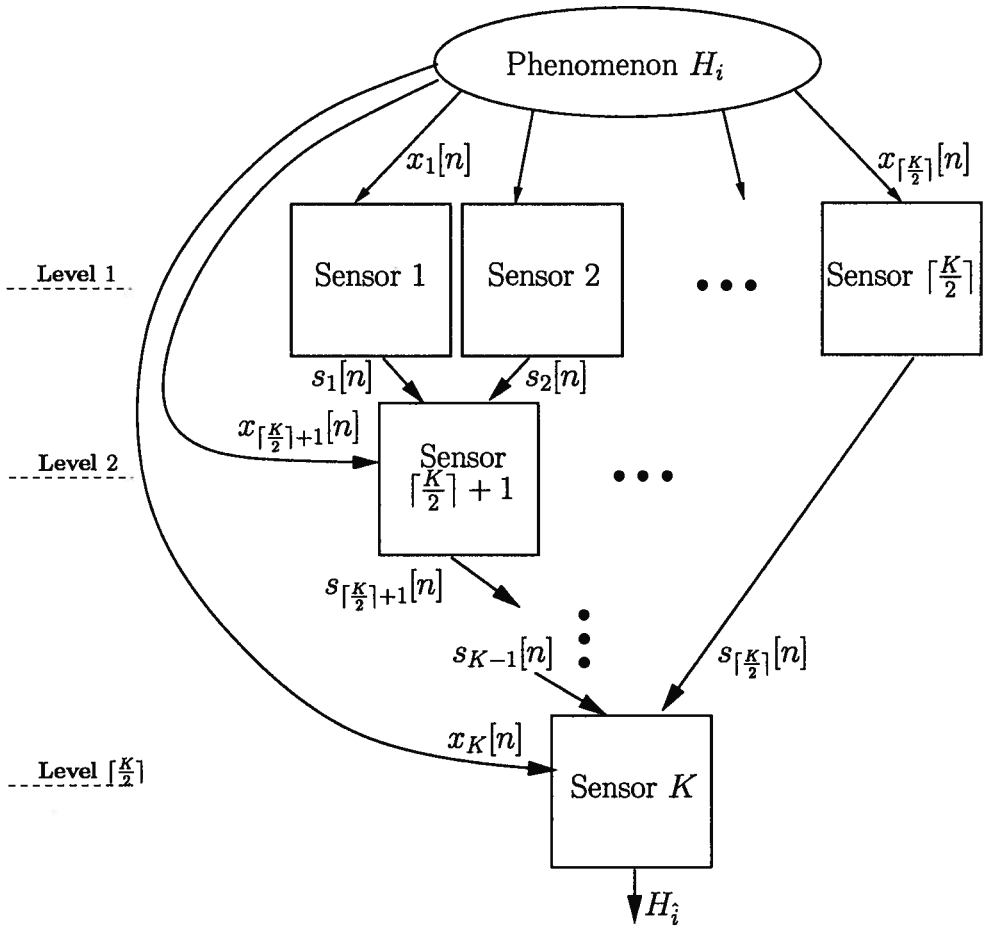

Figure 1.3: Tree topology sensor network.

sensors observe a phenomenon and transmit their decisions, $s_k[n]$, to the next sensor. The remaining $\lfloor \frac{K}{2} \rfloor$ sensors not only make their own observations of the phenomenon, but also receive a decision, $s_k[n]$, from two sensors. Decision fusion is applied and the resultant decision after fusion is transmitted to the next sensor. The global decision is obtained at the $\lceil \frac{K}{2} \rceil$th level of the tree.

1.2 Decision Fusion Literature Review

As mentioned previously, decentralized detection is an important task in WSNs [1, 11, 12, 14]. To limit complexity, the sensors usually make independent decisions based on their respective observations and forward these decisions over

the wireless channel to a FC, which makes a final decision on the hypothesis under test. Most of the existing literature on the decentralized detection problem assumes ideal error–free communication between the sensors and the FC. While this is a reasonable assumption for wired sensors, significant performance degradations may occur if wireless sensors are employed. Therefore, the problem of fusing sensor decisions transmitted over noisy fading channels has received considerable interest recently. For example, channel aware decision fusion for phase–coherent WSNs employing phase–shift keying (PSK) modulation was investigated in [15, 16]. In [17], channel statistics based fusion rules for WSNs employing on/off keying (OOK) modulation were considered. The impact of fading on the performance of power constrained WSNs was studied in [18]. In [19], the performance of type–based multiple access strategies for fading WSNs was analyzed. Furthermore, the problem of optimal power scheduling and decision fusion in fading WSNs with amplify–and–forward processing at the sensors was considered in [20]. Most recently, the impact of channel errors on decentralized detection was studied for PSK, OOK, and frequency–shift keying (FSK) modulation in [21].

Interestingly, existing work on decision fusion for noisy fading channels has mainly considered coherent (e.g., PSK) and noncoherent (e.g., OOK, FSK) modulation schemes. While the former are suitable for static fading channels, the latter are appropriate for extremely fast fading channels, where the fading gain changes from symbol to symbol due to e.g. fast frequency hopping. However, for applications where the fading gains change slowly over time due to the mobility of the sensors and/or FC, noncoherent modulation may not be a preferred choice due to the inherent loss in power efficiency compared to coherent modulation. On the other hand, coherent modulation requires the insertion of pilot symbols for channel estimation, which reduces spectral efficiency and complicates system design. Thus, for conventional point–to–point communication systems, differential PSK (DPSK) is often preferred for sig-

naling over time–varying fading channels [22]. While DPSK does not require instantaneous channel state information (CSI) for detection, the performance loss compared to coherent PSK can be mitigated by multiple–symbol differential detection (MSDD) if statistical CSI is available at the receiver [23–25]. This motivates the investigation of DPSK for transmission in WSNs and the design of corresponding fusion rules.

1.3 Thesis Contribution and Organization

This thesis will consider the parallel topology shown in Fig. 1.1b, which has received the most interest in the literature, and make the following contributions:

1. Formulate the parallel decentralized M–ary hypothesis testing problem in time–variant fading channels.

2. Propose fusion rules that require at most statistical CSI but not any knowledge about instantaneous channel gains.

3. Consider fundamental limits on wireless sensor network performance by investigating performance upper bounds.

4. Estimate probabilities of false alarm and detection for some proposed fusion rules via analytical expressions.

The following chapters will discuss the above contributions in detail. Chapter 2 introduces the system model for a parallel WSN where sensors employ M–ary DPSK (M-DPSK) to report their local decisions to the FC. The chapter will also briefly review MSDD of M–DPSK signals and explain the use of sphere decoding to reduce the complexity of MSDD. The latter concept is referred to as multiple–symbol sphere differential decoding (MSDSD).

In Chapter 3, the M–ary hypothesis testing problem is considered and the optimal multiple–symbol differential (MSD) fusion rule for sensors employing differential encoding is derived. The optimal fusion rule is found to be exponential in both the number of sensors and the observation window size used for MSD decision fusion, and therefore we propose three suboptimal fusion rules with significantly lower complexity and good performance. All proposed fusion rules in this thesis only require statistical CSI but not any knowledge about the instantaneous channel gains.

Chapter 4 considers the special case of ($M = 2$) binary hypothesis testing (BHT) and derives performance bounds valid for the proposed fusion rules. The two performance upper bounds consider the limiting cases, i.e., when the sensors are perfect and when the channel between the sensors and the FC is perfect. Additionally, exact or approximate analytical expressions for the probabilities of false alarm and detection for the suboptimal fusion rules are derived. The analysis allows the system performance to be estimated without resorting to time consuming simulations of the receiver characteristic curve (ROC) for each fusion rule.

In Chapter 5, simulation and, where possible, numerical results are presented for the special case of BHT and the more general case of 4–ary multiple hypothesis testing (MHT). The performance gains achieved by increasing the observation window size, N, of the MSD fusion rules to more than two symbols is shown to be significant. In particular, the performance of coherent detection with perfect knowledge of the channel gains can be approached for large enough observation window sizes. Finally, in Chapter 6 we provide some conclusions for this thesis, as well as several proposals for future work based on the results presented herein.

1.4 Related Publications

- A. Lei and R. Schober. Multiple–Symbol Differential Decision Fusion for Mobile Wireless Sensor Networks. *Submitted to the IEEE Transactions on Wireless Communications*, Mar. 2009.

- A. Lei and R. Schober. Multiple–Symbol Differential Decision Fusion for Mobile Wireless Sensor Networks. *Submitted to IEEE GLOBECOM 2009*, Mar. 2009.

Chapter 2

System Model and Differential Detection

In this chapter, we introduce the WSN system model under consideration. Within the WSN, we will consider the transmission scheme used by the sensors, the fading and noise model used to model the corrupted signal observed by the FC and consider how the signal can be processed by the FC. Moreover, MSDD will also be considered in this chapter, focusing on a lower complexity alternative to MSDD referred to as MSDSD.

2.1 System Model

Consider the parallel distributed multiple hypothesis testing problem where a set $\mathcal{K} \triangleq \{1, 2, \ldots, K\}$ of K sensors are used to decide which one out of M possible hypotheses H_i, $i \in \mathcal{M}$, $\mathcal{M} \triangleq \{0, 1, \ldots, M-1\}$, is present. The *a priori* probability of hypothesis H_i is denoted by $P(H_i)$, $i \in \mathcal{M}$. Fig. 2.1 illustrates the system model which will be discussed in detail in the following subsections.

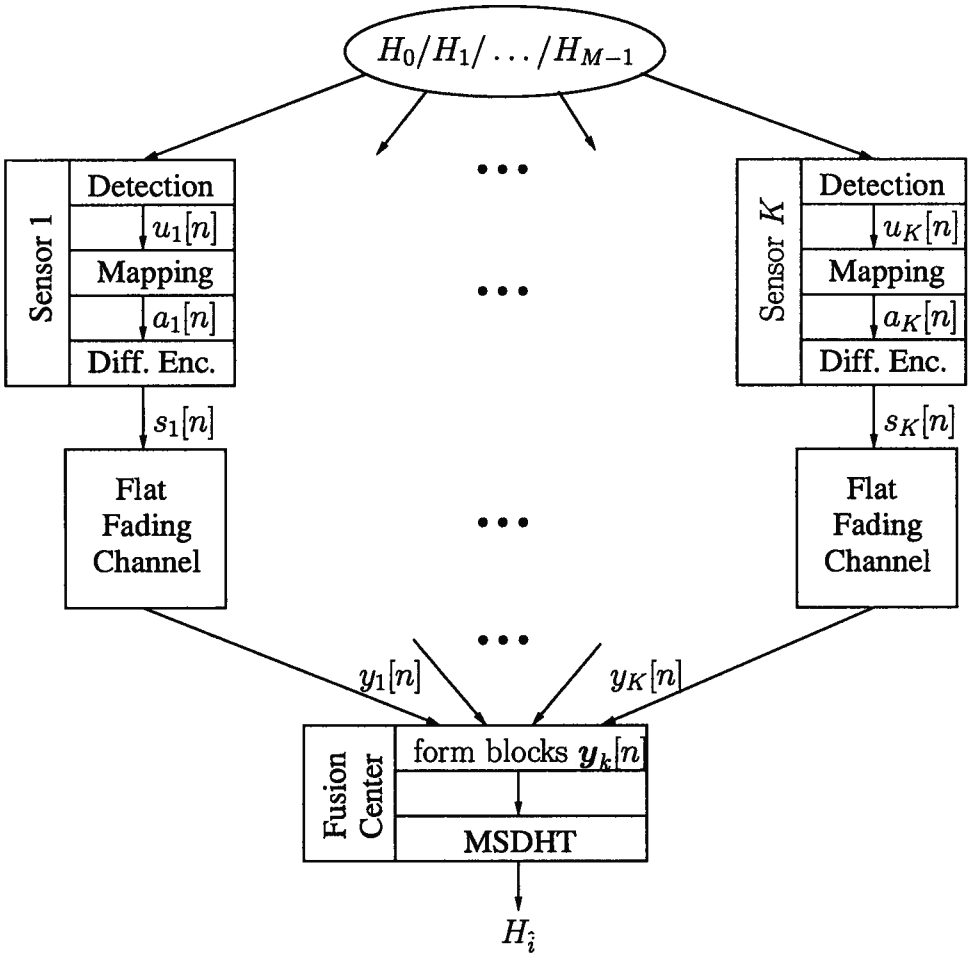

Figure 2.1: System model for MSDHT decision fusion.

2.1.1 Processing at Sensors

At time $n \in \mathbb{Z}$ each sensor $k \in \mathcal{K}$ makes an M–ary decision $u_k[n]$ based on its own noisy observation $x_k[n]$. We assume that the K observations $x_k[n]$, $k \in \mathcal{K}$, are independent of each other, conditioned on the M different hypotheses. The sensors map their local decisions to M–ary PSK (M–PSK) symbols $a_k[n] \in \{w_i | i \in \mathcal{M}\}$, $w_i \triangleq e^{j2\pi i/M}$, such that hypothesis H_i corresponds to the PSK symbol w_i. The differential phase symbols $a_k[n]$ are differentially encoded before transmission over the wireless channel to obtain the absolute phase symbols

$$s_k[n] = a_k[n]s_k[n-1], \tag{2.1}$$

where $s_k[n] \in \{w_i | i \in \mathcal{M}\}$. This differential encoding operation facilitates detection without CSI at the receiver which is particularly useful for transmission over time–variant fading channels [22]. In the context of WSNs, such time–variant channels may arise for example in vehicular WSNs with mobile sensors and/or mobile fusion centers [2], battlefield surveillance [3], or collaborative spectrum sensing with mobile nodes [9]. To keep our model general, we quantify the quality of the local decisions made by the sensors in terms of conditional probabilities $P_k(a_k[n] = w_j | H_i)$, $i \in \mathcal{M}$, $j \in \mathcal{M}$, $k \in \mathcal{K}$.

2.1.2 Channel Model

The sensors communicate with the fusion center over orthogonal flat fading channels using e.g. a time–division multiple access (TDMA) protocol. The received signal from sensor k at time n is given by

$$y_k[n] = \sqrt{\bar{P}_K} h_k[n] s_k[n] + n_k[n], \tag{2.2}$$

where $\bar{P}_K \triangleq \bar{P}/K$ with total transmitted power $\bar{P}$, and $h_k[n]$ and $n_k[n]$ denote the fading gain and zero–mean complex–valued additive white Gaussian noise (AWGN), respectively. The noise is independent, identically distributed (i.i.d.) with respect to both the sensors, k, and time, n, and has variance $\sigma_n^2 \triangleq \mathcal{E}\{|n_k[n]|^2\}$. We assume independent, non–identically distributed (i.n.d.) Rayleigh fading with fading gain variances $\sigma_k^2 \triangleq \mathcal{E}\{|h_k[n]|^2\}$, $k \in \mathcal{K}$. For the temporal correlation of the fading gains, we adopt Clarke's model with $\varphi_{hh,k}[\lambda] \triangleq \mathcal{E}\{h_k[n+\lambda] h_k^*[n]\} = \sigma_k^2 J_0(2\pi B_k T \lambda)$, where B_k denotes the Doppler shift of sensor k and T denotes the time interval between two observations $y_k[n]$ and $y_k[n+1]$. Note that if the sensors use TDMA to report their observations in a round–robin fashion to the fusion center, T is equal to $T = KT_s$, where T_s is the symbol duration. It is also interesting to observe that the effective Doppler shift $B_k T$ increases with decreasing data rate since T increases with decreasing data rate.

2.1.3 Fusion Center Processing

Since the differential encoding operation in (2.1) introduces memory, symbol–by–symbol information fusion is not optimum. Instead, results from point–to–point communication systems suggest that the received signals should be processed on a block–by–block basis [23–25] and will be briefly discussed in the next section. Here, we adopt the same philosophy for information fusion and process blocks of N received signals $\boldsymbol{y}_k \triangleq [y_k[n-N+1]\ y_k[n-N+2]\ \ldots\ y_k[n]]^T$ corresponding to blocks of $N-1$ differential symbols $\boldsymbol{a}_k \triangleq [a_k[n-N+2]\ a_k[n-N+3]\ \ldots\ a_k[n]]^T$, $k \in \mathcal{K}$. Based on these blocks of received signals any one of the $N-1$ differential symbols in $\boldsymbol{a}_k$ can be detected and the corresponding MSD fusion rules will be discussed in the next chapter.

2.2 Differential Detection

The detection of M–DPSK signals in point–to–point links can be accomplished easily by processing blocks of received symbols of length $N = 2$, and considering the phase difference between the current received symbol and the previous received symbol. This detection is known as conventional DD. However, as mentioned previously, if channel statistics are known, MSDD can be used to exploit the memory of the wireless channel thereby improving error performance. Moreover, a low complexity algorithm known as MSDSD can be applied for detection of DPSK signals.

2.2.1 Multiple–Symbol Differential Detection

The problem of MSDD of M–DPSK in additive white Gaussian noise has been addressed by Divsalar *et al.* in [23, 25]. Extensions to fading channels have been considered in [24], noting in the face of time–varying channels, conventional DD is limited by an error floor and the effects can be mitigated by

MSDD. Performance improvement over conventional DD is achieved by jointly detecting blocks of N received symbols simultaneously, thereby exploiting the correlation between phase distortions experienced by neighboring symbols [24, 26]. Adopting the previous notation and ignoring the subscript k for a single user point–to–point system, the received signal at the receiver is

$$\boldsymbol{y} = \sqrt{\bar{P}}\boldsymbol{S}\boldsymbol{h} + \boldsymbol{n}, \tag{2.3}$$

where $\boldsymbol{S} \triangleq \text{diag}\{s[n-N+1]\ s[n-N+2]\ \dots\ s[n]\}$, $\boldsymbol{h} \triangleq [h[n-N+1]\ h[n-N+2]\ \dots\ h[n]]^T$ and $\boldsymbol{n} \triangleq [n[n-N+1]\ n[n-N+2]\ \dots\ n[n]]^T$. With the received vector $\boldsymbol{y}$ being a Gaussian random vector process, the conditional probability density function of $\boldsymbol{y}$, given the blocks of $N-1$ differential symbols $\boldsymbol{a}$, is given by [24]

$$p(\boldsymbol{y}|\boldsymbol{a}) = \frac{1}{\pi^N \det(\boldsymbol{R})} \exp\left(-\boldsymbol{r}^H \boldsymbol{R}^{-1} \boldsymbol{r}\right). \tag{2.4}$$

Here, $\boldsymbol{r} \triangleq [r[n-N+1]\ r[n-N+2]\ \dots\ r[n]]^T$ with $r[n] \triangleq y[n]s^*[n]$ and $\boldsymbol{R} \triangleq \mathcal{E}\{\boldsymbol{r}\boldsymbol{r}^H\} = \bar{P}\boldsymbol{R}_{hh} + \sigma_n^2 \boldsymbol{I}_N$, where $[\boldsymbol{R}_{hh}]_{j,i} = \varphi_{hh}[j-i]$. To simplify our notation, we will address in the following the νth element of vectors $\boldsymbol{y}$ and $\boldsymbol{a}$ by $y(\nu)$, $1 \le \nu \le N$, and $a(\nu)$, $1 \le \nu \le N-1$, respectively. The maximum–likelihood (ML) decision for $\boldsymbol{a}$ requires maximizing (2.4). Eliminating all irrelevant terms, this is equivalent to maximizing

$$\hat{\boldsymbol{a}} = \underset{\boldsymbol{a}}{\text{argmax}}\ \left\{-\boldsymbol{r}^H \boldsymbol{R}^{-1} \boldsymbol{r}\right\} = \underset{\boldsymbol{a}}{\text{argmax}}\ \left\{2\Re\left\{\sum_{\mu=1}^{N}\sum_{\nu=\mu+1}^{N} t_{\mu\nu} y(\mu) y^*(\nu) \prod_{\xi=\mu}^{\nu-1} a(\xi)\right\}\right\}, \tag{2.5}$$

where $t_{\mu\nu} \triangleq -[\boldsymbol{R}^{-1}]_{\mu,\nu}$. If blocks of received signals are properly processed, performance improves as the block size $N \ge 2$ increases and approaches the performance of coherent detection for $N \to \infty$ [23, 26]. Applying (2.5) requires computing the argument for each candidate vector $\boldsymbol{a}$, i.e., computational complexity grows exponentially with N. To reduce the computation complexity of this problem, MSDSD was proposed in [27].

2.2.2 Multiple–Symbol Differential Sphere Decoding

The purpose of adopting sphere decoding is to reduce the number of candidate vectors $\boldsymbol{a}$ considered without excluding the ML solution. Considering again (2.5), maximization with respect to $\boldsymbol{a}$ is equivalent to maximization with respect to the absolute phase symbols $\boldsymbol{s} \triangleq [s(1)\ s(2)\ \ldots\ s(N)]^T$ given that the phase ambiguity is fixed, e.g., $s(N) = 1$. The ML decision is equivalent to

$$\hat{\boldsymbol{s}} = \underset{\boldsymbol{s},\, s(N)=1}{\operatorname{argmin}} \left\{ \boldsymbol{s}^H \boldsymbol{U}^H \boldsymbol{U} \boldsymbol{s} \right\} = \underset{\boldsymbol{s},\, s(N)=1}{\operatorname{argmin}} \left\{ \|\boldsymbol{U}\boldsymbol{s}\|^2 \right\}, \tag{2.6}$$

where $\boldsymbol{U} \triangleq (\boldsymbol{L}^H \operatorname{diag}\{\boldsymbol{y}\})^*$ is an upper triangular matrix and $\boldsymbol{L}$ is a lower triangular matrix obtained from the Cholesky factorization of $\boldsymbol{R}^{-1} \triangleq \boldsymbol{L}\boldsymbol{L}^H$ [27]. The sphere decoder considers all candidate vector $\boldsymbol{s}$ that lie within a sphere of radius R,

$$\boldsymbol{s}^H \boldsymbol{U}^H \boldsymbol{U} \boldsymbol{s} \leq R^2. \tag{2.7}$$

Since $\boldsymbol{U}$ is an upper triangular matrix, $\|\boldsymbol{U}\boldsymbol{s}\|^2$ can be partially computed given some elements within $\boldsymbol{s}$ are fixed. Assuming (preliminary) decisions $\hat{s}(l)$ for the last $N - l$ components $s(l)$, $\nu + 1 \leq l \leq N$, we define the squared length

$$d_{\nu+1}^2 = \sum_{l=\nu+1}^{N} \left| \sum_{\mu=l}^{N} u_{l\mu} \hat{s}(\mu) \right|^2. \tag{2.8}$$

For possible values of $s(\nu)$, the condition

$$d_\nu^2 = \left| u_{\nu\nu} s(\nu) + \sum_{\mu=\nu+1}^{N} u_{\nu\mu} \hat{s}(\mu) \right|^2 + d_{\nu+1}^2 \leq R^2, \tag{2.9}$$

must be satisfied, where the computation of (2.9) uses the partially computed squared length $d_{\nu+1}^2$ accounting for the squared length of the last $N - l$ components. For any $s(\nu)$ with $d_\nu^2 > R^2$, any symbol $s(l)$ for $1 \leq l \leq \nu - 1$ will also have $d_l^2 > R^2$ and therefore can be eliminated early in the search process. Once a vector $\hat{\boldsymbol{s}}$ is found to satisfy (2.7), the radius is updated to reflect the new radius, $R^2 = \|\boldsymbol{U}\hat{\boldsymbol{s}}\|^2$. This is repeated until no vector $\boldsymbol{s}$ is within a sphere

of radius R and the ML solution $\hat{\boldsymbol{s}}$ is obtained. The algorithm is shown in [27, Fig. 1].

Fig. 2.2 shows a binary tree to illustrate the search for $\hat{\boldsymbol{s}}$ in a 2–DPSK system with $N = 4$. The root node of the tree is a node without parents and represents the initial search, i.e., when $s(N) = 1$, to fix the phase ambiguity. Any nodes in the tree excluding the nodes at the top and bottom level of the tree can be referred to as either a parent or a child since these nodes spawn from nodes at a higher level of the tree and the nodes themselves can spawn M child nodes at a lower level. The last level ($\nu = 1$) of the tree consists of only leaf nodes, nodes that do not have any children. M^{N-1} leaf nodes represent the number of possible candidate vector $\boldsymbol{s}$. In Fig. 2.2, the vectors $[s(N-1), \ldots s(\nu+1)\, s(\nu)]^T$ are provided beside each node. The following illustrates a search for the ML solution.

1. Starting at the root node, the Schnoor–Euchner (SE) search strategy is applied to estimate the node with the minimum incremental length, i.e., $\left|u_{\nu\nu}s(\nu) + \sum_{\mu=\nu+1}^{N} u_{\nu\mu}\hat{s}(\mu)\right|^2$, and moves to that node [28]. This is repeated until a candidate $\hat{\boldsymbol{s}}$ is found, and the radius is updated. The initial search always produces one $\hat{\boldsymbol{s}}$ if $R \to \infty$. As a result of the SE algorithm, $M-1$ nodes can always be eliminated at the bottom level of the tree ($\nu = 1$). Fig. 2.2a shows a possible path traversed by the sphere decoder.

2. Once a candidate $\hat{\boldsymbol{s}}$ is found, the search is repeated by moving up one level ($\nu = 2$) and $M-1$ other child nodes spanned by the same parent are considered. We want to consider the next best child node that has the shortest incremental length other than the current $\hat{s}(\nu)$. This is shown in Fig. 2.2b. The squared length, d_ν, at this new node is computed using (2.9) and compared with R.

3. In Fig 2.2c, it is assumed that d_ν at this new node computed in step 2

is greater than R. Since (2.9) is not satisfied, all child nodes that are spawned from this node can be eliminated and they are not a possible ML solution. Furthermore, it is known from step 2 that the considered node has the next smallest incremental distance compared to $\hat{s}(\nu)$. As a result, the remaining $M-2$ nodes considered in step 2 can also be eliminated since they, like the current node, would also not satisfy (2.9). Thus, we must move up one level ($\nu = 3$) and consider again child nodes that are spawned from the same parent.

4. Fig. 2.2d shows a new $\hat{\boldsymbol{s}}$ satisfying (2.7). This new $\hat{\boldsymbol{s}}$ is selected as the current best ML solution and the radius is updated. The previous $\hat{s}$ can then be eliminated as the ML solution.

5. Repeating step 2, another node must be compared to consider other candidate $\hat{\boldsymbol{s}}$ that satisfy (2.7) to ensure no $\boldsymbol{s}$ is excluded, otherwise the solution obtained by the SD maybe sub–optimal. Fig. 2.2e shows that there are still 2 possible candidates.

6. Fig. 2.2f assumes that the squared length computed in the previous step does not satisfy (2.9) and therefore no other candidate $\boldsymbol{s}$ is left and $\hat{\boldsymbol{s}}$ obtained from step 4 is the ML solution.

As illustrated in the above example, SD can reduce the complexity of MSDD by eliminating unnecessary comparisons. However, as will be shown later, the number of eliminations will depend on the channel signal–to-noise ratio (SNR). At low SNR the complexity of MSDSD is still quite high but decreases rapidly as channel SNR increases. Although methods such as early terminate [29] and K–best [30] algorithms have been proposed as a means to reduce complexity by finding a sub–optimal solution, we will only consider the case where $R \rightarrow \infty$ and the search terminates only when the ML solution is obtained.

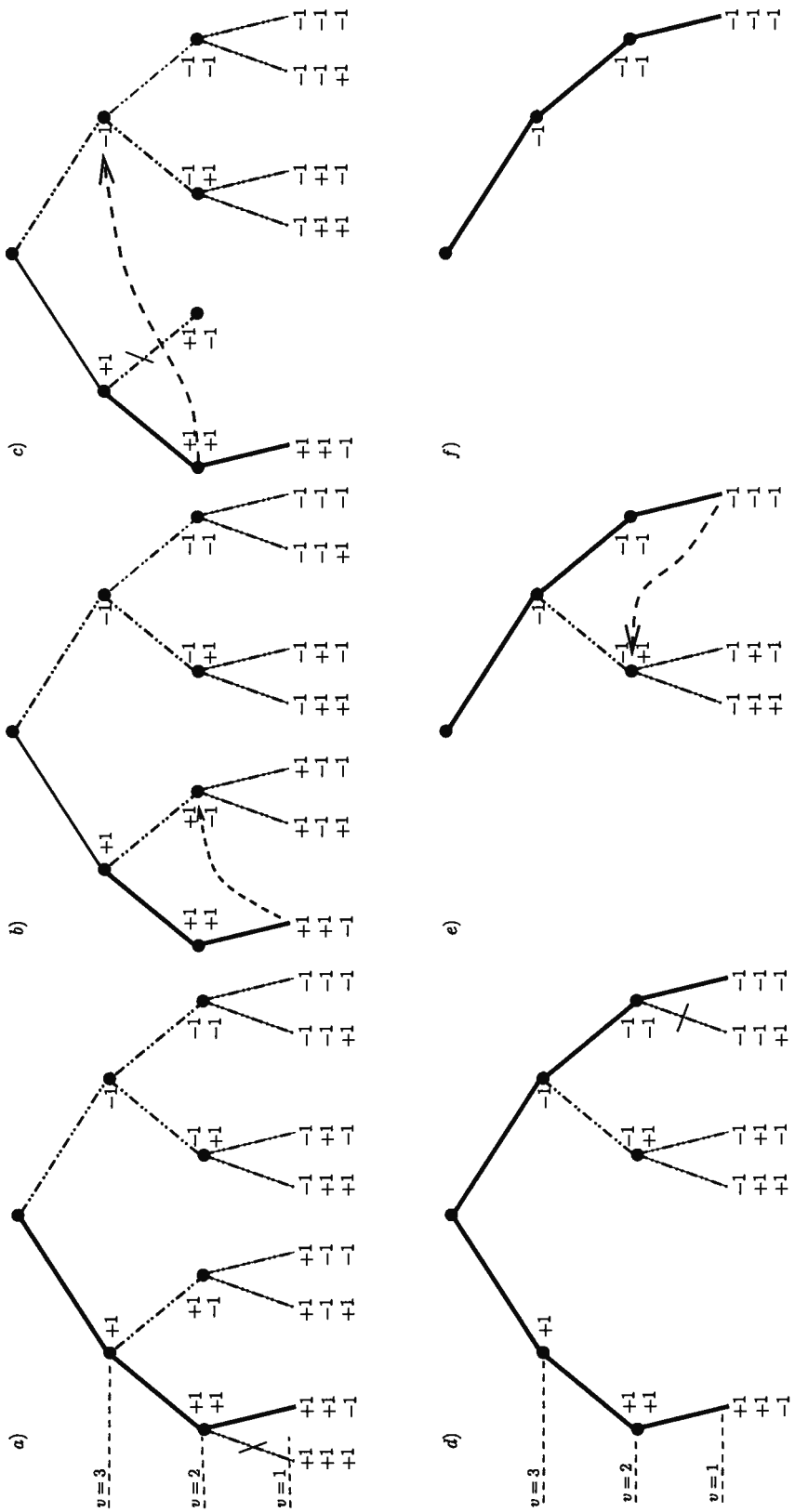

Figure 2.2: Sphere decoding illustration for differential block of $N = 4$.

Chapter 3

Multiple–Symbol Differential Decision Fusion

In this chapter, the optimal and several suboptimal fusion rules for the system model introduced in Chapter 2 will be derivied. For this derivation, it will be assumed that the fusion center has knowledge of both the statistical properties of the channel and the performance index $P_k(a_k[n] = w_j|H_i)$, $i \in \mathcal{M}$, $j \in \mathcal{M}$, of the sensors $k \in \mathcal{K}$. However, as will become clear in the following, for some of the considered fusion rules one or both of these conditions can be relaxed. We denote the indices of the differential symbols considered for detection by ν_0, $\nu_0 \in \{1, 2, \ldots, N-1\}$. To simplify the notation, we will drop the index of the differential symbol considered for detection wherever possible and denote it by $a_k = a_k(\nu_0)$.

3.1 Optimal Fusion Rule

The optimal fusion rule based on the observations $\boldsymbol{y} \triangleq [\boldsymbol{y}_1^T\ \boldsymbol{y}_2^T\ \ldots\ \boldsymbol{y}_K^T]^T$ can be formulated as

$$\hat{H}_i = \underset{H_i, i \in \mathcal{M}}{\operatorname{argmax}} \{\log(P(H_i|\boldsymbol{y})) + \alpha_i\}, \tag{3.1}$$

where α_i is a bias term which allows the prioritization of certain hypotheses. A bias may be useful for example in applications such as spectrum sensing for a cognitive radio where a missed detection is less desirable than a false alarm. Since we assume that fading and noise are independent across different sensors, the conditional probability $P(H_i|\boldsymbol{y})$ can be rewritten as

$$P(H_i|\boldsymbol{y}) = \frac{p(\boldsymbol{y}|H_i)P(H_i)}{p(\boldsymbol{y})} = \prod_{k=1}^{K} \frac{p_k(\boldsymbol{y}_k|H_i)P(H_i)}{p_k(\boldsymbol{y}_k)}. \tag{3.2}$$

Furthermore, the conditional pdf $p_k(\boldsymbol{y}_k|H_i)$ of sensor k can be expanded as

$$\begin{aligned} p_k(\boldsymbol{y}_k|H_i) &= \sum_{j=0}^{M-1} p_k(\boldsymbol{y}_k|a_k = w_j)P_k(a_k = w_j|H_i) \\ &= \frac{1}{M^{N-2}} \sum_{j=0}^{M-1} \sum_{\boldsymbol{a}_k \in \mathcal{A}_j} p_k(\boldsymbol{y}_k|\boldsymbol{a}_k)P_k(a_k = w_j|H_i), \end{aligned} \tag{3.3}$$

where $\mathcal{A}_j$ contains all M^{N-2} possible vectors $\boldsymbol{a}_k$ with $a_k = w_j$ and the conditional pdf $p_k(\boldsymbol{y}_k|\boldsymbol{a}_k)$ is given by (2.4). Combining (2.4) and (3.1)–(3.3), and omitting all irrelevant terms will yield the optimal MSD fusion rule

$$\begin{aligned} \hat{H_i} &= \underset{H_i\, i\in\mathcal{M}}{\operatorname{argmax}} \left\{ \sum_{k=1}^{K} \log \left(\sum_{j=0}^{M-1} \sum_{\boldsymbol{a}_k \in \mathcal{A}_j} p_k(\boldsymbol{y}_k|\boldsymbol{a}_k)P_k(a_k = w_j|H_i) \right) + \beta_i \right\} \\ &= \underset{H_i,\, i\in\mathcal{M}}{\operatorname{argmax}} \left\{ \sum_{k=1}^{K} \log \left(\sum_{j=0}^{M-1} \sum_{\boldsymbol{a}_k \in \mathcal{A}_j} \exp \left(2\Re \left\{ \sum_{\mu=1}^{N} \sum_{\nu=\mu+1}^{N} t_{\mu\nu}^{k} y_k(\mu) y_k^*(\nu) \right.\right.\right.\right. \\ &\qquad \left.\left.\left.\left. \times \prod_{\xi=\mu}^{\nu-1} a_k(\xi) \right\} \right) P_k(a_k = w_j|H_i) \right) + \beta_i \right\}, \end{aligned} \tag{3.4}$$

where $\beta_i \triangleq \alpha_i + \log(P(H_i))$ denotes the new bias term.

Discussion: Despite its optimal performance, the MSD fusion rule in (3.4) has several short–comings: (a) The complexity of the fusion rule in (3.4) is exponential in both K and N; (b) Because of the large dynamic range of the exponential functions in (3.4), especially for high channel SNRs (i.e., $\bar{P}_K \sigma_k^2/\sigma_n^2 \gg 1$), the optimal fusion rule causes numerical problems, especially

in fixed point implementations; and (c) The optimal fusion rule requires statistical CSI (in the form of $t_{\mu\nu}^k$) and knowledge of the sensor performance (in form of $P_k(a_k = w_j|H_i)$), though we note that the coefficients $t_{\mu\nu}^k$ are related to the coefficients of a linear predictor for the process $r_k[n]$ and can be efficiently computed using adaptive algorithms [31, 32]. In light of the above–listed drawbacks of the optimal fusion rule, there is a need to search for suboptimal fusion rules that do not have these drawbacks but still provide a good performance.

3.2 Chair–Varshney (CV) Fusion Rule

The complexity of the optimal fusion rule can be significantly reduced by assuming that the double sum on the right hand side (RHS) of (3.3) is dominated by the ML vector $\hat{\boldsymbol{a}}_k \triangleq [\hat{a}_k(1) \ldots \hat{a}_k(N-1)]^T$ which maximizes $p_k(\boldsymbol{y}_k|\boldsymbol{a}_k)$, i.e., $p_k(\boldsymbol{y}_k|\hat{\boldsymbol{a}}_k) \gg p_k(\boldsymbol{y}_k|\boldsymbol{a}_k)$, $\boldsymbol{a}_k \neq \hat{\boldsymbol{a}}_k$, $k \in \mathcal{K}$. This is a valid assumption for high channel SNR. In this case, the optimal fusion rule can be simplified to

$$H_{\hat{i}} = \underset{H_i, i\in\mathcal{M}}{\operatorname{argmax}} \left\{ \sum_{k=1}^{K} \log(p_k(\boldsymbol{y}_k|\hat{\boldsymbol{a}}_k) P_k(\hat{a}_k|H_i)) + \beta_i \right\}, \tag{3.5}$$

where $\hat{a}_k = \hat{a}_k(\nu_0)$ denotes the element of $\hat{\boldsymbol{a}}_k$ which is considered for detection. We note that the ML vectors $\hat{\boldsymbol{a}}_k$, $k \in \mathcal{K}$, can be efficiently obtained from $\boldsymbol{y}_k$ by applying the MSDSD algorithm in [27, Fig. 1], cf. Section 2.2.2. For binary hypothesis testing ($M = 2$), (3.5) can be expressed as a likelihood ratio

$$\Lambda_{\text{cv}} = \sum_{\hat{a}_k=1} \log \frac{P_k(\hat{a}_k|H_1)}{P_k(\hat{a}_k|H_0)} + \sum_{\hat{a}_k=-1} \log \frac{P_k(\hat{a}_k|H_1)}{P_k(\hat{a}_k|H_0)}, \tag{3.6}$$

and we decide in favor of H_1 if Λ_{cv} exceeds threshold $\gamma_0 \triangleq \beta_0 - \beta_1$, and for H_0 otherwise. Thus, (3.5) and (3.6) can be regarded as the MSD version of the familiar CV fusion rule [11].

Discussion: The complexity of the suboptimal fusion rules in (3.5) and (3.6) grows only linearly with the increase in the number of sensors K. Furthermore, for sufficiently high channel SNR the average complexity of MSDSD

is polynomial in N [27], and thus, the complexity of the proposed fusion rule is also polynomial in N. Similar to the optimal fusion rule, knowledge of the sensor performance and, for $N > 2$, the statistical CSI, is required for the CV fusion rule. For $N = 2$, based on (3.5) it can be shown that statistical CSI is not required if the channels are i.i.d. (i.e., $\boldsymbol{R}_k = \boldsymbol{R}$, $\forall k$).

3.3 Fusion Rule for Ideal Local Sensors (ILS)

For derivation of the CV fusion rule it was implicitly assumed that the uncertainty about the hypothesis at the fusion center originated only from the local sensor decisions, whereas the channel between the sensors and the fusion center was assumed ideal. The opposite assumptions, however, can also be made, namely that the local sensor decisions are ideal, i.e., $P_k(a_k = w_i|H_i) = 1$ and $P_k(a_k = w_j|H_i) = 0$ for $j \neq i$, and the uncertainty at the fusion center is due to the noisy transmission channel only. In this case, $\boldsymbol{a}_k = \boldsymbol{a}$, $\forall k \in \mathcal{K}$, is valid and the optimal ML block decision rule for $\boldsymbol{a}$ is given by

$$\hat{\boldsymbol{a}} = \underset{\boldsymbol{a}}{\operatorname{argmax}} \left\{ \sum_{k=1}^{K} \log(p_k(\boldsymbol{y}_k|\boldsymbol{a})) + \beta_i \right\}, \tag{3.7}$$

where the bias β_i is determined by the trial symbol $a = a(\nu_0) = w_i$, $i \in \mathcal{M}$, and the hypothesis estimate $H_{\hat{i}}$ can be directly obtained from the relevant element $\hat{a} = \hat{a}(\nu_0) = w_{\hat{i}}$ of $\hat{\boldsymbol{a}}$. For the binary case, it is convenient to express (3.7) in terms of a likelihood ratio

$$\Lambda_{\text{ILS}} = \sum_{k=1}^{K} \log \left(\frac{p_k(\boldsymbol{y}_k|\hat{\boldsymbol{a}}^1)}{p_k(\boldsymbol{y}_k|\hat{\boldsymbol{a}}^0)} \right), \tag{3.8}$$

where $\hat{\boldsymbol{a}}^j$ is that vector $\boldsymbol{a} \in \mathcal{A}_j$ which maximizes $\sum_{k=1}^{K} \log(p_k(\boldsymbol{y}_k|\boldsymbol{a}))$. In particular, $H_{\hat{i}} = H_1$ is chosen if $\Lambda_{\text{ILS}} > \gamma_0 = \beta_0 - \beta_1$, and $H_{\hat{i}} = H_0$ otherwise. The computational complexity of the fusion rules in (3.7) and (3.8) is only linear in K but still exponential in N if a brute force search for all possible $\boldsymbol{a}$ is conducted. Similar to the CV fusion rule, the application of sphere decoding

is the key to reducing complexity further. For this purpose, we rewrite (3.7) as

$$\hat{\boldsymbol{s}} = \underset{\boldsymbol{s},\, s(N)=1}{\operatorname{argmin}} \left\{ \sum_{k=1}^{K} \boldsymbol{s}^H \boldsymbol{U}_k^H \boldsymbol{U}_k \boldsymbol{s} - \beta_i \right\}. \tag{3.9}$$

$\boldsymbol{s} \triangleq [s(1)\ s(2)\ \ldots\ s(N)]^T$ contains the absolute phase symbols from which the elements of $\boldsymbol{a}$ are obtained as $a(j) = s(j+1)s^*(j)$. Because of the phase ambiguity inherent in (3.9), we can set $s(N) = 1$ without loss of generality. The sum over k and the bias term β_i in (3.9) make a direct application of the MSDSD algorithm in [27] impossible. However, as will be explained in the following, a modified version of MSDSD can be used to solve (3.9) efficiently provided that $\beta_i \leq 0$, $i \in \mathcal{M}$. The latter condition can always be fulfilled by properly choosing α_i, $i \in \mathcal{M}$.

The modified MSDSD only examines candidate vectors that meet

$$\sum_{k=1}^{K} \boldsymbol{s}^H \boldsymbol{U}_k^H \boldsymbol{U}_k \boldsymbol{s} - \beta_i \leq R^2. \tag{3.10}$$

Assuming we have found (preliminary) decisions $\hat{s}(l)$ for the last $N-l$ components $s(l)$, $\nu+1 \leq l \leq N$, we can define an equivalent squared length

$$d_{\nu+1}^2 = \sum_{l=\nu+1}^{N} \sum_{k=1}^{K} \left| \sum_{\mu=l}^{N} u_{l\mu}^k \hat{s}(\mu) \right|^2 - \beta_{\hat{i}}\, \delta[\nu_0 - \nu - 1], \tag{3.11}$$

where $u_{\nu\mu}^k \triangleq [\boldsymbol{U}_k]_{\nu,\mu}$ and $\beta_{\hat{i}}$ is obtained from $\hat{s}(\nu_0+1)\hat{s}^*(\nu_0) = \hat{a} = w_{\hat{i}}$. Comparing (3.10) and (3.11), possible values for $s(\nu)$ have to satisfy

$$d_\nu^2 = \sum_{k=1}^{K} \left| u_{\nu\nu}^k s(\nu) + \sum_{\mu=\nu+1}^{N} u_{\nu\mu}^k \hat{s}(\mu) \right|^2 - \beta_i\, \delta[\nu_0 - \nu] + d_{\nu+1}^2 \leq R^2, \tag{3.12}$$

where β_i is determined by $\hat{s}(\nu_0+1)s^*(\nu_0) = a = w_i$. Similar to the MSDSD in [27], once a valid vector $\hat{\boldsymbol{s}}$ is found, i.e., $\nu = 1$ is reached, the radius R is dynamically updated by $R := d_1$, and the search is repeated starting with $\nu = 2$ and the new radius R. If condition (3.12) cannot be met for some index ν, ν is incremented and another value of $s(\nu)$ is tested. Based on (3.10)–(3.12)

the modified MSDSD algorithm given by Algorithms 1 and 2 can be obtained. Consequently, with an initial radius of $R \to \infty$, the algorithm will always find a solution to (3.7).

Discussion: The complexity of the ILS fusion rule is linear in K and for high channel SNR polynomial in N. While statistical CSI is still required for $N > 2$, the local sensor performance $P_k(a_k = w_j|H_i)$, $i \in \mathcal{M}$, $j \in \mathcal{M}$, does not have to be known at the fusion center. Of course, the price to be paid for this advantage is a loss in performance compared to the optimal fusion rule. For $N = 2$, it can be shown based on (3.7) that statistical CSI is not required if the channels are i.i.d.

Algorithm 1 Pseudocode for MSDSD for ILS Fusion Rule

function $[\hat{s}, d^2] = \text{MSDSD-ILS}(U_1, U_2, \ldots, U_K, M, N, R, \nu_0, \beta)$ ▸ initial radius R, vector of bias term β
  $s_N := 1$ ▸ fix last component of s
  $d_N^2 := \sum_{k=1}^{K} |u_{NN}^k|^2$ ▸ initialize squared length
  $\nu := N-1$ ▸ start at component $\nu = N-1$
  for $k = 1$ to K **do** $q_\nu^k := \sum_{l=\nu+1}^{N} u_{wl}^k s_l$ **end for** ▸ sum the last $N-\nu$ components
  $[\text{step}_\nu, n_\nu] = \text{findBestILS}(q_\nu^1, \ldots, q_\nu^K, u_{\nu\nu}^1, \ldots, u_{\nu\nu}^K, M)$ ▸ find with 1^{st} candidate for comp. at $\nu = N-1$
  search := 1
  while search = 1 **do**
    $d_\nu^2 := \sum_{k=1}^{K} |u_{\nu\nu}^k e^{\frac{j2\pi}{M}\text{step}_\nu(n_\nu)} + q_\nu^k|^2 + d_{\nu+1}^2$ ▸ update squared length
    if $\nu = \nu_0$ **then**
      $i := \text{mod}\{M + (\text{step}_{\nu+1}(n_{\nu+1}) - \text{step}_\nu(n_\nu), M\}$ ▸ find current estimated hypothesis $\alpha(\nu_0) = w_i$
      $d_\nu^2 := d_\nu^2 - \beta_i$ ▸ bias length due to hypothesis i being the candidate
    end if
    if $d_\nu < R$ **and** $n_\nu \leq M$ **then** ▸ check search radius and constellation size
      $s_\nu := e^{\frac{j2\pi}{M}\text{step}_\nu(n_\nu)}$ ▸ store candidate component
      if $\nu \neq 1$ **then** ▸ check if last component is reached
        $\nu := \nu - 1$ ▸ move down to next component
        for $k = 1$ to K **do** $q_\nu^k := \sum_{l=\nu+1}^{N} u_{wl}^k s_l$ **end for** ▸ sum the last $N-\nu$ components
        $[\text{step}_\nu, n_\nu] = \text{findBestILS}(q_\nu^1, \ldots, q_\nu^K, u_{\nu\nu}^1, \ldots, u_{\nu\nu}^K, M)$ ▸ find the 1^{st} candidate for component at ν
      else ▸ first component reached
        $\hat{s} := s$ ▸ store best estimated sequence so far
        $R := d_\nu$ ▸ update sphere radius
        $\nu := \nu + 1$ ▸ move up to previous component
        while $n_\nu = M$ **and** search = 1 **do** ▸ move up until $n_\nu \neq M$
          if $\nu = N-1$ **then** search := 0 **else** $\nu := \nu + 1$ **end if** ▸ terminate search if $\nu = N-1$ else move up to prev. comp.
        end while
        $n_\nu := n_\nu + 1$ ▸ count examined candidates for component at ν
      end if
    else
      if $\nu = N-1$ **then**
        search := 0 ▸ terminate search if $\nu = N-1$
      else
        $\nu := \nu + 1$ ▸ move up to previous component
        while $n_\nu = M$ **and** search = 1 **do** ▸ move up until $n_\nu \neq M$
          if $\nu = N-1$ **then** search := 0 **else** $\nu := \nu + 1$ **end if** ▸ terminate search if $\nu = N-1$ else move up to prev. comp.
        end while
      end if
      $n_\nu =: n_\nu + 1$ ▸ count examined candidates for component at ν
    end if
  end while
end function

Algorithm 2 Pseudocode for findBest used in MSDSD

function $[\text{STEP}_\nu, n_\nu] = \text{findBestILS}(q_\nu^1, \ldots, q_\nu^K, u_{\nu\nu}^1, \ldots, u_{\nu\nu}^K, M)$ ▸ find order of M–PSK signal point to test according to SE strategy
  for $m = 0$ to $M-1$ **do** $d_{test} := \sum_{k=1}^{K} |u_{\nu\nu}^k e^{\frac{j2\pi}{M}m} + q_\nu^k|^2$ **end for** ▸ Compute all possible square length contributions
  $\text{step}_\nu := sort\_ascending(d_{test})$ ▸ returns step_ν which contains values m sorted in order of increasing value of d_{test}
  $n_\nu := 1$ ▸ counter of examined candidates for component ν
end function

3.4 Max–Log Fusion Rule

For high channel SNR (i.e., $\bar{P}_K \sigma_k^2/\sigma_n^2 \gg 1$, $\forall k \in \mathcal{K}$) one of the exponentials in (3.4) will be dominant and the max–log approximation, which is well known from the Turbo–coding literature [33], can be applied

$$\begin{aligned} H_{\hat{i}} &= \underset{H_i, i\in\mathcal{M}}{\operatorname{argmax}} \left\{ \sum_{k=1}^{K} \max_{j\in\mathcal{M}, \boldsymbol{a}_k\in\mathcal{A}_j} \{\log(p_k(\boldsymbol{y}_k|\boldsymbol{a}_k)) + \log(P_k(a_k = w_j|H_i))\} + \beta_i \right\} \\ &= \underset{H_i, i\in\mathcal{M}}{\operatorname{argmax}} \left\{ \sum_{k=1}^{K} \max_{j\in\mathcal{M}, \boldsymbol{a}_k\in\mathcal{A}_j} \left\{ 2\Re\left\{ \sum_{\mu=1}^{N} \sum_{\nu=\mu+1}^{N} t_{\mu\nu}^k y_k(\mu) y_k^*(\nu) \prod_{\xi=\nu}^{\mu-1} a_k(\xi) \right\} \right. \right. \\ &\qquad \left. \left. + \log(P_k(a_k = w_j|H_i)) \right\} + \beta_i \right\}. \end{aligned} \tag{3.13}$$

The max–log fusion rule in (3.13) is computationally more efficient and numerically more stable than the optimal fusion rule in (3.4) since exponential functions are avoided in (3.13). However, if (3.13) is implemented in a straightforward fashion, its computational complexity is still exponential in N, since for every test hypothesis H_i, $i \in \mathcal{M}$, the maximum of $\log(p_k(\boldsymbol{y}_k|\boldsymbol{a}_k))$ has to be found over all $\boldsymbol{a}_k \in \mathcal{A}_j$, $j \in \mathcal{M}$. However, the max–log fusion rule can be rewritten as

$$H_{\hat{i}} = \underset{H_i, i\in\mathcal{M}}{\operatorname{argmax}} \left\{ \sum_{k=1}^{K} \left(\max_{j\in\mathcal{M}} \left\{ \log(p_k(\boldsymbol{y}_k|\hat{\boldsymbol{a}}_k^j)) + \log(P_k(\hat{a}_k = w_j|H_i)) \right\} \right) + \beta_i \right\}, \tag{3.14}$$

where $\hat{\boldsymbol{a}}_k^j$ is that $\boldsymbol{a}_k \in \mathcal{A}_j$ which maximizes $p_k(\boldsymbol{y}_k|\boldsymbol{a}_k)$. $\hat{\boldsymbol{a}}_k^j$ can be efficiently computed using sphere decoding. In particular, one simple method to obtain $\hat{\boldsymbol{a}}_k^j$, $j \in \mathcal{M}$, is by slightly modifying the MSDSD algorithm in [27, Fig. 1] by forcing $a(\nu_0) = w_j$ even when this does not produce the ML solution. This modified MSDSD must be applied M times and will be referred to as repeated tree search (RTS) MSDSD. However, a more efficient method can be obtained by noting that in RTS–MSDSD there are a large number of redundant squared length computations. This is avoided if the SD obtains all $\hat{\boldsymbol{a}}_k^j$, $j \in \mathcal{M}$, in a single search. We will refer to this as single tree search (STS)

MSDSD. The STS–MSDSD approach was proposed in [34] by introducing multiple radii for comparisons rather than just one radius representing the current best estimate of $\hat{\boldsymbol{s}}$. Applying STS–MSDSD, we consider M different radii, $\vec{R} \triangleq [R_0\ R_2\ \ldots\ R_{M-1}]$, correspond to M different $\hat{\boldsymbol{a}}_k^j$. Similar to the MSDSD algorithm in [27, Fig. 1], STS–MSDSD only examines candidate vector that meet the condition

$$\boldsymbol{s}_k^H \boldsymbol{U}_k^H \boldsymbol{U}_k \boldsymbol{s}_k \leq R_j^2, \tag{3.15}$$

where R_j is determined by $s_k(\nu_0+1)s_k^*(\nu_0) = w_j$. Similarly, possible values of $s_k(\nu)$ have to satisfy both

$$d_\nu^2 = \left| u_{\nu\nu}^k s_k(\nu) + \sum_{\mu=\nu+1}^{N} u_{\nu\mu}^k \hat{s}_k(\mu) \right|^2 + d_{\nu+1}^2 \leq R_{\max}^2, \quad \nu_0 < \nu \leq M+1 \tag{3.16}$$

$$d_\nu^2 = \left| u_{\nu\nu}^k s_k(\nu) + \sum_{\mu=\nu+1}^{N} u_{\nu\mu}^k \hat{s}_k(\mu) \right|^2 + d_{\nu+1}^2 \leq R_j^2, \quad 1 \leq \nu \leq \nu_0 \tag{3.17}$$

where $R_{\max} = \max_j R_j$. The condition (3.16) is necessary to ensure that when $\boldsymbol{a}_k^j$ is not the ML solution, it does not get eliminated early on in the search. Once a valid $\hat{\boldsymbol{s}}_k$ is obtained, the radius R_j is dynamically updated depending on $\hat{s}(\nu+1)\hat{s}(\nu)$. The pseudocode for this STS–MSDSD is given in Algorithm 3, where the functions findBest() and findNext() are given in [27, Fig. 1]. For the binary case, $M = 2$, (3.14) is equivalent to choosing H_1 if likelihood ratio $\Lambda_{\text{m-log}}$ exceeds threshold $\gamma_0 = \beta_0 - \beta_1$, and H_0 otherwise, where $\Lambda_{\text{m-log}}$ is defined as

$$\Lambda_{\text{m-log}} = \sum_{k=1}^{K} \max_{j\in\mathcal{M}} \min_{i\in\mathcal{M}} \left\{ \log \left(\frac{p_k(\boldsymbol{y}_k|\hat{\boldsymbol{a}}_k^j) P_k(\hat{a}_k = w_j|H_1)}{p_k(\boldsymbol{y}_k|\hat{\boldsymbol{a}}_k^i) P_k(\hat{a}_k = w_i|H_0)} \right) \right\}. \tag{3.18}$$

Discussion: The complexity of the max–log fusion rule is linear in K and for high channel SNR polynomial in N. The implementation of (3.14) and (3.18) requires knowledge of the sensor performance $P_k(a_k = w_j|H_i)$, $i \in \mathcal{M}$, $j \in \mathcal{M}$. Furthermore, the complexity of the max–log fusion rule is higher than that of the CV and ILS fusion rules discussed in Sections 3.2 and 3.3,

respectively. For the max–log fusion rule MK vectors $\hat{\boldsymbol{a}}_k^j$, $j \in \mathcal{M}$, $k \in \mathcal{K}$, have to be obtained per decision. In contrast, for the CV and ILS fusion rules, only K and one ML vectors have to be computed, respectively. In addition, in contrast to the CV and ILS fusion rules, the max–log fusion rule requires statistical CSI even for $N = 2$ and i.i.d. channels. On the other hand, the max–log fusion rule achieves a superior performance compared to the CV and ILS fusion rules, cf. Chapter 5.

Algorithm 3 Pseudocode for MSDSD for Max–Log Fusion Rule

```
function [ŝ^g] = MSDSD-ILS(U, M, N, R̄, ν0)                    ▸ initial vector of radius R̄,
    s_N := 1                                                   ▸ fix last comp. of s
    d_N^2 := | u_NN |^2                                        ▸ initialize squared length
    ν := N − 1                                                 ▸ start at comp. ν = N − 1
    q_ν := Σ_{l=ν+1}^{N} u_νl s_l                              ▸ sum the last N − ν components
    R_max := max{R_0, ..., R_{M−1}}                            ▸ set maximum radius
    [m_ν, step_ν, n_ν] = findBest(q_ν, u_νν, M)                ▸ find best candidate for comp. at ν = N − 1
    search := 1
    while search = 1 do
        d_ν^2 := |u_νν e^{j2π/M m_ν} + q_ν|^2 + d_{ν+1}^2      ▸ update squared length
        if d_ν < R_max and n_ν ≤ M then                        ▸ check maximum radius and constellation size
            s_ν := e^{j2π/M n_ν}                               ▸ store candidate comp.
            if ν == ν0 then g := mod[M + (m_{ν+1} − m_ν), M] else i := 1 end if   ▸ check which radius to compare
            if ν == 1 and d_ν < R_g then                       ▸ check if last comp. reached and radius R_g
                ŝ^g := s                                       ▸ store best estimated sequence so far
                R_g := d_ν                                     ▸ update radius R_g
                R_max := max {R_0, ..., R_{M−1}}               ▸ update maximum radius
                ν := ν + 1                                     ▸ move up to previous comp.
                while n_ν = M and search = 1 do                ▸ move up until n_ν ≠ M
                    if ν = N − 1 then search := 0 else ν := ν + 1 end if   ▸ terminate search if ν = N − 1 else move up to prev. comp.
                end while
                [m_ν, step_ν, n_ν] = findNext(m_ν, step_ν, n_ν)   ▸ find next component at ν to examine
            else if (ν > ν0 or ( ν ≤ ν0 and d_ν < R_g)) and n_ν ≤ M then   ▸ check if need to compare radius R_g and constellation size
                ν := ν − 1                                     ▸ move down to next comp.
                q_ν := Σ_{l=ν+1}^{N} u_νl s_l                  ▸ sum the last N − ν components
                [m_ν, step_ν, n_ν] = findBest(q_ν, u_νν, M)    ▸ find best candidate for comp. at ν
            else
                if ν = N − 1 then
                    search := 0                                ▸ terminate search if ν = N − 1
                else
                    if ν ≠ ν0 then ν := ν + 1 end if           ▸ force to compare other s(ν) comp. if ν = ν0, else move up
                    while n_ν = M and search = 1 do            ▸ move up until n_ν ≠ M
                        if ν = N − 1 then search := 0 else ν := ν + 1 end if   ▸ terminate search if ν = N − 1 else move up to prev. comp.
                    end while
                end if
                [m_ν, step_ν, n_ν] = findNext(m_ν, step_ν, n_ν)   ▸ find next component at ν to examine
            end if
        else
            if ν = N − 1 then
                search := 0                                    ▸ terminate search if ν = N − 1
            else
                ν := ν + 1                                     ▸ move up to previous compo.
                while n_ν = M and search = 1 do                ▸ move up until n_ν ≠ M
                    if ν = N − 1 then search := 0 else ν := ν + 1 end if   ▸ terminate search if ν = N − 1 else move up to prev. comp.
                end while
            end if
            [m_ν, step_ν, n_ν] = findNext(m_ν, step_ν, n_ν)    ▸ find next component at ν to examine
        end if
    end while
end function
```

Chapter 4

Analysis of Suboptimal Fusion Rules

An analysis of the optimal fusion rule does not seem to be possible. Therefore, we concentrate in this section on the CV, ILS, and max–log fusion rules and on general performance bounds valid for any fusion rule. To make the analysis tractable, we assume $M = 2$, i.e., $\mathcal{M} = \{0, 1\}$, i.i.d. channels, i.e., $\sigma_k^2 = \sigma^2$, $B_k = B$, $\boldsymbol{R}_k = \boldsymbol{R}$, $t_{\mu\nu}^k = t_{\mu\nu}$, and $p_k(\boldsymbol{y}_k|\boldsymbol{a}_k) = p(\boldsymbol{y}_k|\boldsymbol{a}_k)$, $\forall k \in \mathcal{K}$, and identical sensors with probability of false alarm $P_f \triangleq P(a_k = -1|H_0) = P_k(a_k = -1|H_0)$ and probability of detection $P_d \triangleq P(a_k = -1|H_1) = P_k(a_k = -1|H_1)$, $\forall k \in \mathcal{K}$.

4.1 Performance Bounds

Before considering specific fusion rules, we provide two performance upper bounds valid for any fusion rule including the optimal one.

1) Bound I: For the first bound, we assume that all sensors make correct decisions and decision errors at the fusion center are due to transmission errors only, i.e., $\boldsymbol{a}_k = \boldsymbol{a}$, $\forall k \in \mathcal{K}$, and zero bias, i.e., $\beta_0 = \beta_1 = 0$. In this case, the sensor network is equivalent to a point–to–point transmission with K–fold receive diversity and conventional MSDD [24, 25] is the optimal fusion rule.

Thus, the probabilities of false alarm and detection are given by

$$P_{f_0} = \mathrm{BER}_{\nu_0} \quad \text{and} \quad P_{d_0} = 1 - \mathrm{BER}_{\nu_0}, \tag{4.1}$$

where BER_{ν_0} denotes the probability that $a = a(\nu_0)$ was transmitted and $\hat{a} \neq a$, $\hat{a} \in \{\pm 1\}$, $1 \leq \nu_0 \leq N-1$, was detected, i.e., BER_{ν_0} is the bit error rate (BER) for 2–DPSK symbol $a(\nu_0)$ for point–to–point transmission and MSDD at the receiver. BER_{ν_0} can be lower bounded as [26]

$$\mathrm{BER}_{\nu_0} \geq (\mathrm{PEP}_{\nu_0} + \mathrm{PEP}_{\nu_0+1})/2, \quad 1 \leq \nu_0 \leq N-1, \tag{4.2}$$

where the pairwise error probability (PEP), PEP_{ν_0}, is the probability that vector $\boldsymbol{s}$ was transmitted and $\hat{\boldsymbol{s}}(\nu_0) \triangleq [s(1) \ldots s(\nu_0 - 1)\, \hat{s}(\nu_0)\, s(\nu_0 + 1) \ldots s(N)]^T$, $\hat{s}(\nu_0) \neq s(\nu_0)$, was detected. The averaging over two error events in (4.2) is necessary, since, because of the differential encoding, $\hat{a}(\nu_0) \neq a(\nu_0)$ may be caused by either $\hat{\boldsymbol{s}}(\nu_0)$ or $\hat{\boldsymbol{s}}(\nu_0 + 1)$. Note that in order to get performance upper bounds, we only count error events causing a single erroneous symbol, $\hat{s}(\nu) \neq s(\nu)$, in (4.2). Taking into account the K–fold diversity, we obtain from [26, Eq. (12a)] for the PEP for the problem at hand

$$\mathrm{PEP}_\nu = \left[\frac{1}{2}\left(1 - \frac{1}{\sqrt{1 + 1/\rho_\nu}}\right)\right]^K \sum_{k=0}^{K-1} \frac{1}{2^k} \binom{K+k-1}{K-1} \left(1 + \frac{1}{\sqrt{1 + \frac{2}{2+\rho_\nu}}}\right)^k, \tag{4.3}$$

where $\rho_\nu \triangleq -t_{\nu\nu}(\bar{P}_K + \sigma_n^2) - 1$, $1 \leq \nu \leq N$. Eqs. (4.1)–(4.3) constitute a performance upper bound for any fusion rule with noisy sensors. This bound becomes tight for optimal decision fusion if transmission errors dominate the overall performance, which is the case for example at low channel SNRs and for highly reliable local sensors.

2) Bound II: For the second bound, we assume a noise–free transmission channel, i.e., the decision errors at the fusion center are caused by local decision errors at the sensors only. In this case, the CV fusion rule is optimum and the

corresponding probabilities of false alarm and detection are given by [11]

$$P_{f_0} = \sum_{i=K_{\gamma_0}}^{K} \binom{K}{i} P_f^i (1-P_f)^{K-i} \text{ and } P_{d_0} = \sum_{i=K_{\gamma_0}}^{K} \binom{K}{i} P_d^i (1-P_d)^{K-i}, \quad (4.4)$$

where K_{γ_0}, $0 \leq K_{\gamma_0} \leq K$, depends on decision threshold γ_0 as follows

$$K_{\gamma_0} = \left\lceil \frac{\gamma_0 - K \log\left(\frac{1-P_d}{1-P_f}\right)}{\log \frac{P_d(1-P_f)}{P_f(1-P_d)}} \right\rceil . \quad (4.5)$$

For realistic, noisy transmission channels, (4.4) constitutes a performance upper bound which becomes tight for high channel SNRs.

4.2 CV Fusion Rule

Considering (3.6) the probabilities of false alarm and detection at the fusion center can be expressed as

$$P_{f_0} = \sum_{i=K_{\gamma_0}}^{K} \binom{K}{i} P_0^i (1-P_0)^{K-i} \text{ and } P_{d_0} = \sum_{i=K_{\gamma_0}}^{K} \binom{K}{i} P_1^i (1-P_1)^{K-i}, \quad (4.6)$$

where $P_0 = P(H_{\hat{i}} = H_1 | H_0)$ and $P_1 = P(H_{\hat{i}} = H_1 | H_1)$. P_0 and P_1 can be expanded as

$$\begin{aligned} P_i &= P(H_{\hat{i}} = H_1 | a_k = -1) P(a_k = -1 | H_i) + P(H_{\hat{i}} = H_1 | a_k = 1) P(a_k = 1 | H_i) \\ &= (1 - \text{BER}_{\nu_0}) P_{\text{x}_i} + \text{BER}_{\nu_0} (1 - P_{\text{x}_i}), \end{aligned} \quad (4.7)$$

where $\text{x}_0 = f$ and $\text{x}_1 = d$, and BER_{ν_0} is the BER of 2–DPSK for a point–to–point link without diversity and MSDD at the receiver. This BER can be approximated as [26, Table I, Eq. (12)]

$$\text{BER}_{\nu_0} \approx \text{PEP}_{\nu_0} + \text{PEP}_{\nu_0+1}, \quad 1 \leq \nu_0 \leq N-1, \quad (4.8)$$

$$\text{PEP}_{\nu} = \frac{1}{2}\left(1 - \frac{1}{\sqrt{1+1/\rho_\nu}}\right). \quad (4.9)$$

For the special case of $N = 2$, (4.8) is exact, while it is an accurate approximation for $N > 2$ and sufficiently high channel SNR. Using (4.6)–(4.9) the

probabilities of false alarm and detection for the CV decision rule can be computed approximately (exactly) for $N > 2$ ($N = 2$).

4.3 ILS Fusion Rule

The ILS decision in (3.7) is influenced by the local sensor decisions $a_k(\nu)$, $1 \leq k \leq K$, $1 \leq \nu \leq N-1$, which makes an exact analysis for $N > 2$ intractable and renders both approximations and bounds loose. Therefore, in this subsection, we concentrate on the case $N = 2$. The probabilities of false alarm and detection can be expressed as

$$P_{f_0} = \sum_{\bar{\boldsymbol{a}}} P(\hat{a} = -1|\bar{\boldsymbol{a}})P(\bar{\boldsymbol{a}}|H_0) \quad \text{and} \quad P_{d_0} = \sum_{\bar{\boldsymbol{a}}} P(\hat{a} = -1|\bar{\boldsymbol{a}})P(\bar{\boldsymbol{a}}|H_1), \tag{4.10}$$

where $P(\hat{a} = -1|\bar{\boldsymbol{a}})$ denotes the probability that the fusion center detects $\hat{a} = -1$, i.e., $H_{\hat{i}} = H_1$, given the local sensor decisions $\bar{\boldsymbol{a}} \triangleq [a_1\ a_2\ \ldots\ a_K]^T$. Furthermore, the conditional sensor decision probabilities in (4.10) are given by $P(\bar{\boldsymbol{a}}|H_0) = P_f^{K-k_0}(1-P_f)^{k_0}$ and $P(\bar{\boldsymbol{a}}|H_1) = P_d^{K-k_0}(1-P_d)^{k_0}$, where k_0 denotes the number of elements of $\bar{\boldsymbol{a}}$ that are equal to 1. Based on (3.8) $P(\hat{a} = -1|\bar{\boldsymbol{a}})$ can be expressed as $P(\hat{a} = -1|\bar{\boldsymbol{a}}) = \Pr\{-\Lambda_{\text{ILS}} < -\gamma_0\}$, which leads to

$$P(\hat{a} = -1|\bar{\boldsymbol{a}}) = \frac{1}{2\pi j}\int_{c-j\infty}^{c+j\infty} \Phi_{\text{ILS}}(s|\bar{\boldsymbol{a}})e^{-\gamma_0 s}\,\frac{\mathrm{d}s}{s}, \tag{4.11}$$

where $\Phi_{\text{ILS}}(s|\bar{\boldsymbol{a}})$ denotes the Laplace transform of the pdf of $-\Lambda_{\text{ILS}}$ given $\bar{\boldsymbol{a}}$ and $\hat{a}_k = \hat{a}$, and c is a small positive constant that lies in the region of convergence of the integral. Closer examination of (3.8) for $N = 2$ reveals that Λ_{ILS} is a

quadratic form of Gaussian random variables, and can be expressed as

$$\begin{aligned}\Lambda_{\text{ILS}} &= -4\sum_{k=1}^{K}\Re\{t_{12}^k y_k(1)y_k^*(2)\} = -2\sum_{k=1}^{K}(t_{12}^k y_k(1)y_k^*(2)+t_{21}^k y_k^*(1)y_k(2)) \\ &= -2\sum_{k=1}^{K}(t_{12}^k a_1(\sqrt{\bar{P}_K}h_k(1)+n_k(1))(\sqrt{\bar{P}_K}h_k(2)+n_k(2))^* \\ &\quad +t_{21}^k a_k(\sqrt{\bar{P}_K}h_k(1)+n_k(1))^*(\sqrt{\bar{P}_K}h_k(2)+n_k(2))) \\ &= \boldsymbol{u}\boldsymbol{M}(\bar{\boldsymbol{a}})\boldsymbol{u}, \end{aligned} \tag{4.12}$$

where $\boldsymbol{u} \triangleq [\sqrt{\bar{P}_K}h_1[n]+n_1[n],\ \sqrt{\bar{P}_K}h_1[n-1]+n_1[n-1],\ \ldots,\ \sqrt{\bar{P}_K}h_K[n-1]+n_K[n-1]]^T$ and $\boldsymbol{M}(\bar{\boldsymbol{a}}) \triangleq \text{diag}\{\boldsymbol{M}_1(a_1),\ldots,\boldsymbol{M}_K(a_K)\}$ with $\boldsymbol{M}_k(a_k) \triangleq 2a_k\begin{bmatrix}0 & t_{12}\\ t_{21} & 0\end{bmatrix}$. Consequently, we obtain

$$\Phi_{\text{ILS}}(s|\bar{\boldsymbol{a}}) = 1/\det(\boldsymbol{I}_{2K}+s\bar{\boldsymbol{R}}\boldsymbol{M}(\bar{\boldsymbol{a}})), \tag{4.13}$$

where $\bar{\boldsymbol{R}} \triangleq \boldsymbol{I}_K \otimes \boldsymbol{R}$. We note that the integral in (4.11) can be numerically evaluated efficiently using e.g. Gauss–Chebyshev quadrature rules [35]

$$\begin{aligned}\Pr\{-\Lambda_{\text{ILS}} < -\gamma_0\} &= \frac{1}{2\pi j}\int_{c-j\infty}^{c+j\infty}\Phi_{\text{ILS}}(s|\bar{\boldsymbol{a}})e^{-\gamma_0 s}\frac{ds}{s} \\ &\approx \frac{1}{2\bar{N}}\Re\{\sum_{i=1}^{\bar{N}}\Phi_{\text{ILS}}(c+c\delta_i|\bar{\boldsymbol{a}})[1-j\delta_i]e^{c\gamma_0(1+j\delta_i)}\},\end{aligned} \tag{4.14}$$

where $\Phi_{\text{ILS}}(s|\bar{\boldsymbol{a}})$ is given in (4.13) with $s = c+c\delta_i$ and $\delta_i = \tan(\frac{2i-1}{2\bar{N}}\pi)$. The selection of $\bar{N}$ and c affects the accuracy of the numerical method, with c selected to satisfy $0 < c < c_0$, where c_0 denotes the smallest real part of all singularities of $\Phi_{\text{ILS}}(s|\bar{\boldsymbol{a}})$, and $\bar{N}$ selected as large as possible to achieve a desired accuracy. Thus, the exact probabilities of false alarm and detection for the ILS fusion rule with $N = 2$ can be efficiently computed from (4.10)–(4.11), (4.13)–(4.14).

4.4 Max–Log Fusion Rule

For the max–log fusion rule, the probabilities of false alarm and detection can be expressed as

$$P_{f_0} = \Pr\{-\Lambda_{\mathrm{m-log}} < -\gamma_0 | H_0\} \quad \text{and} \quad P_{d_0} = \Pr\{-\Lambda_{\mathrm{m-log}} < -\gamma_0 | H_1\}, \quad (4.15)$$

cf. (3.18). Denoting the Laplace transform of the pdf of the negative log–likelihood ratio $-\Lambda_{\mathrm{m-log}}$ by $\Phi_{\mathrm{m-log}}(s|H_i)$, (4.15) can be rewritten as

$$P_{\mathrm{x}_i} = \frac{1}{2\pi j} \int_{c-j\infty}^{c+j\infty} \Phi_{\mathrm{m-log}}(s|H_i) e^{-\gamma_0 s} \frac{\mathrm{d}s}{s}, \quad i \in \mathcal{M}, \quad (4.16)$$

where $\mathrm{x}_0 = f_0$ and $\mathrm{x}_1 = d_0$. Since P_{f_0} and P_{d_0} can be obtained by numerical integration from (4.16) if $\Phi_{\mathrm{m-log}}(s|H_i)$ is known, the remainder of this section will be devoted to calculation of this Laplace transform. As the fading gains and noise samples in the different diversity branches are i.i.d., respectively, $\Phi_{\mathrm{m-log}}(s|H_i)$ can be expressed as

$$\Phi_{\mathrm{m-log}}(s|H_i) = (\Phi_z(s|H_i))^K, \quad i \in \mathcal{M}, \quad (4.17)$$

where $\Phi_z(s|H_i)$ denotes the Laplace transform of the pdf of

$$z_k \triangleq \max_{j\in\mathcal{M}} \min_{i\in\mathcal{M}} \left\{ \log \left(\frac{p(\boldsymbol{y}_k|\hat{\boldsymbol{a}}_k^j) P(\hat{a}_k = w_j|H_0)}{p(\boldsymbol{y}_k|\hat{\boldsymbol{a}}_k^i) P(\hat{a}_k = w_i|H_1)} \right) \right\}. \quad (4.18)$$

$\Phi_z(s|H_i)$ can be rewritten as

$$\Phi_z(s|H_i) = (1 - P_{\mathrm{x}_i})\Phi_z(s|\hat{a} = -1, a = 1) + P_{\mathrm{x}_i}\Phi_z(s|\hat{a} = -1, a = -1), \; i \in \mathcal{M}, \quad (4.19)$$

where $\mathrm{x}_0 = f$ and $\mathrm{x}_1 = d$, and $\Phi_z(s|\hat{a}, a)$ denotes the Laplace transform of the pdf of z_k given $a_k = a$ and $\hat{a}$. For calculation of $\Phi_z(s|\hat{a}, a)$ it is useful to note that for $M = 2$, (4.18) can be rewritten as

$$z_k = \log \left(\frac{\max\{p(\boldsymbol{y}_k|\hat{\boldsymbol{a}}_k^0)(1 - P_f), p(\boldsymbol{y}_k|\hat{\boldsymbol{a}}_k^1)P_f\}}{\max\{p(\boldsymbol{y}_k|\hat{\boldsymbol{a}}_k^0)(1 - P_d), p(\boldsymbol{y}_k|\hat{\boldsymbol{a}}_k^1)P_d\}} \right). \quad (4.20)$$

Using the definition $y \triangleq \log(p(\boldsymbol{y}_k|\hat{\boldsymbol{a}}_k^0)/p(\boldsymbol{y}_k|\hat{\boldsymbol{a}}_k^1))|_{a=1}$ and assuming $P_f < 0.5$ and $P_d > 0.5$, we can show that (4.20) can be rewritten as

$$z_k = \begin{cases} \beta_1, & ay < b_1 \\ ay + \beta_2, & b_1 \leq ay \leq b_2 \\ \beta_3, & ay > b_2 \end{cases}, \tag{4.21}$$

where $\beta_1 \triangleq \log(P_f/P_d)$, $\beta_2 \triangleq \log((1-P_f)/P_d)$, $\beta_3 \triangleq \log((1-P_f)/(1-P_d))$, $b_1 \triangleq \log(P_f/(1-P_f))$, and $b_2 \triangleq \log(P_d/(1-P_d))$. To arrive at (4.21) for $a = -1$, we have exploited

$$\log(p(\boldsymbol{y}_k|\hat{\boldsymbol{a}}_k^0)/p(\boldsymbol{y}_k|\hat{\boldsymbol{a}}_k^1))|_{a=1} = -\log(p(\boldsymbol{y}_k|\hat{\boldsymbol{a}}_k^0)/p(\boldsymbol{y}_k|\hat{\boldsymbol{a}}_k^1))|_{a=-1}. \tag{4.22}$$

For convenience (4.21) is illustrated in Fig. 4.1 for the case $a = 1$. Fig. 4.1 reveals that the max–log fusion rule soft–limits the log–likelihood ratios y of the individual sensors at the fusion center by taking into account the *a priori* values P_f and P_d. It is interesting to note that Fig. 4.1 can also be related to

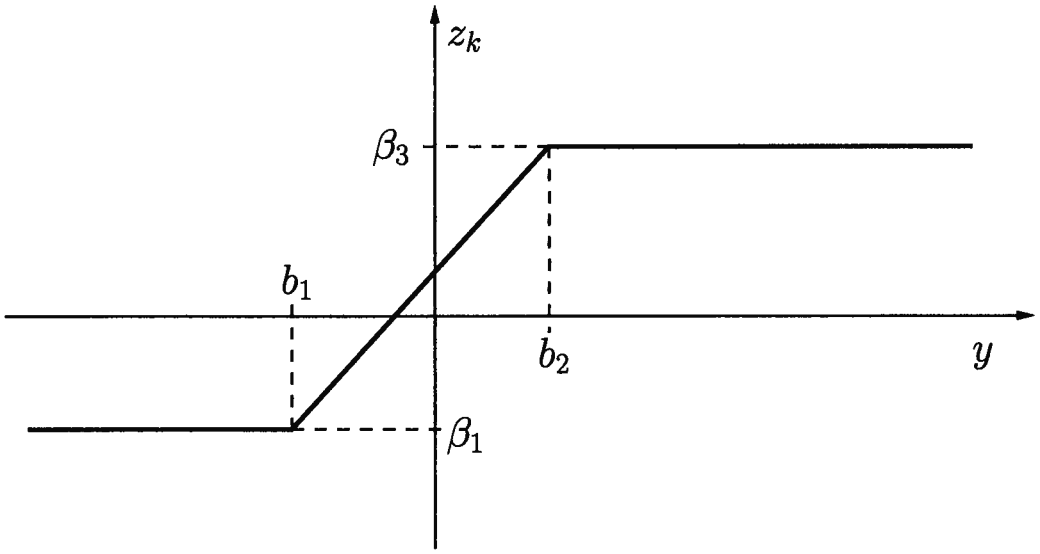

Figure 4.1: Illustration of relationship between z_k and y for $a = 1$. Note that with the definitions in Section 4.4, $b_1 < 0$, $b_2 > 0$, $\beta_1 < 0$, and $\beta_3 > 0$ as long as $P_d > 0.5$ and $P_f < 0.5$.

the CV fusion rule where $b_1 = b_2 = 0$ and $a = 1$

$$z_k = \begin{cases} \beta_1, & y < 0 \\ \beta_3, & y > 0 \end{cases} . \tag{4.23}$$

By denoting the pdf of y by $p_y(y)$ and exploiting (4.21), we can express $\Phi_z(s|\hat{a} = -1, a)$ as

$$\Phi_z(s|\hat{a} = -1, a) = \int_{-\infty}^{b_1} e^{-s\beta_1} p_y(ay)\,\mathrm{d}y + \int_{b_1}^{b_2} e^{-s(y+\beta_2)} p_y(ay)\,\mathrm{d}y + \int_{b_2}^{\infty} e^{-s\beta_3} p_y(ay)\,\mathrm{d}y. \tag{4.24}$$

For calculation of $p_y(y)$, we distinguish in the following the cases $N = 2$ and $N > 2$.

$N = 2$: For $N = 2$, the only possible error event leading to $\hat{a} = -1$ is $\hat{\boldsymbol{s}} = [-s(1)\ 1]^T$ when $\boldsymbol{s} = [s(1)\ 1]^T$ and y can be expressed as $y = -\boldsymbol{r}_k^H \boldsymbol{R}_k^{-1} \boldsymbol{r}_k|_{\boldsymbol{s}} + \boldsymbol{r}_k^H \boldsymbol{R}_k^{-1} \boldsymbol{r}_k|_{\hat{\boldsymbol{s}}}$. In other words, y is simply the decision variable for conventional differential detection. Thus, the Laplace transform of $p_y(y)$ is given by [26, Eq. (27)]

$$\Phi_y(s) = \frac{-v_1 v_2}{(s + v_1)(s - v_2)} \tag{4.25}$$

where $v_{1|2} = (\sqrt{1 + 1/\rho_{\nu_0}} \mp 1)/2$ with $\nu_0 = 1$. From (4.25) we can calculate $p_y(y)$ as

$$p_y(y) = \frac{v_1 v_2}{v_1 + v_2} \left(e^{-v_1 y} \bar{u}(y) + e^{v_2 y} \bar{u}(-y) \right). \tag{4.26}$$

Combining (4.24) and (4.26) we obtain

$$\begin{aligned} \Phi_z(s|\hat{a} = -1, a) \;=\; & \frac{v_i v_j}{v_i + v_j} \left(\frac{e^{-s\beta_1 + v_j b_1}}{v_j} + e^{-s\beta_2} \left(\frac{1 - e^{(-s+v_j)b_1}}{-s + v_j} \right.\right. \\ & \left.\left. + \frac{1 - e^{-(s+v_i)b_2}}{s + v_i} \right) + \frac{e^{-s\beta_3 - v_i b_2}}{v_i} \right), \end{aligned} \tag{4.27}$$

where $(i, j) = (1, 2)$ and $(i, j) = (2, 1)$ for $a = 1$ and $a = -1$, respectively.

$N > 2$: For $N > 2$ the problem is more difficult, since there are more than one possible error events that lead to $\hat{a}(\nu_0) = \hat{a} = -1$. The most likely error events are $\hat{\boldsymbol{s}}_{\nu_0}$ and $\hat{\boldsymbol{s}}_{\nu_0+1}$ which differ from $\boldsymbol{s}$ only in positions ν_0 and

$\nu_0 + 1$, respectively. The corresponding likelihood ratios are denoted by $y_1 \triangleq \log(p(\boldsymbol{y}_k|\hat{\boldsymbol{a}}_k^0)/p(\boldsymbol{y}_k|\hat{\boldsymbol{a}}_k^1))|_{\hat{\boldsymbol{s}}_{\nu_0}}$ and $y_2 \triangleq \log(p(\boldsymbol{y}_k|\hat{\boldsymbol{a}}_k^0)/p(\boldsymbol{y}_k|\hat{\boldsymbol{a}}_k^1))|_{\hat{\boldsymbol{s}}_{\nu_0+1}}$. To make the problem tractable, we assume that $\hat{\boldsymbol{s}}_{\nu_0}$ and $\hat{\boldsymbol{s}}_{\nu_0+1}$ are the only relevant error events, i.e., we neglect all other error events, which is a valid approximation for high channel SNRs. In this case, y is given by $y = \min\{y_1, y_2\}$. In order to get closed–form results, we make the following two additional approximations: (a) y_1 and y_2 are independent and (b) y_1 and y_2 are identically distributed. Both assumptions are justified for high channel SNRs. By exploiting results from order statistics [36] and [26], we obtain for the pdf of y

$$p_y(y) = 2p_{y_1}(y)(1 - P_{y_1}(y)), \tag{4.28}$$

where $p_{y_1}(y) = \frac{v_1 v_2}{v_1+v_2}\left(e^{-v_1 y}\bar{u}(y) + e^{v_2 y}\bar{u}(-y)\right)$, cf. (4.26), $P_{y_1}(y) = \int_{-\infty}^{y} p_{y_1}(x)\,\mathrm{d}x$, and $v_{1|2} = (\sqrt{1 + 1/\rho_{\nu_0}} \mp 1)/2$ with $1 \leq \nu_0 \leq N-1$. Combining (4.24) and (4.28) leads to the following closed–form expression for $\Phi_z(s|\hat{a} = -1, a = 1)$ and $\Phi_z(s|\hat{a} = -1, a = 1)$

$$\begin{aligned}\Phi_z(s|\hat{a} = -1, a = 1) \;=\; & 2\left(\frac{v_1 v_2}{v_1 + v_2}\right)^2 \left(e^{-s\beta_1}\left(\frac{(v_1+v_2)e^{v_2 b_1}}{v_1 v_2^2} - \frac{e^{2v_2 b_1}}{2v_2^2}\right)\right. \\ & + e^{-s\beta_2}\left(\frac{(v_1+v_2)(e^{-(s-v_2)b_1} - 1)}{v_1 v_2 (s - v_2)} + \frac{1 - e^{-(s-2v_2)b_1}}{v_2(s-2v_2)}\right. \\ & \left.\left. + \frac{1 - e^{-(s+2v_1)b_2}}{v_1(s+2v_1)}\right) + \frac{e^{-s\beta_3 - 2v_1 b_2}}{2v_1^2}\right),\end{aligned} \tag{4.29}$$

$$\begin{aligned}\Phi_z(s|\hat{a} = -1, a = -1) \;=\; & 2\left(\frac{v_1 v_2}{v_1 + v_2}\right)^2 \left(e^{-s\beta_3}\left(\frac{(v_1+v_2)e^{-v_2 b_2}}{v_1 v_2^2} - \frac{e^{-2v_2 b_2}}{2v_2^2}\right)\right. \\ & + e^{-s\beta_2}\left(\frac{(v_1+v_2)(1 - e^{-(s+v_2)b_2})}{v_1 v_2 (s + v_2)} + \frac{e^{-(s+2v_2)b_1} - 1}{v_2(s+2v_2)}\right. \\ & \left.\left. + \frac{e^{-(s-2v_1)b_1} - 1}{v_1(s-2v_1)}\right) + \frac{e^{-s\beta_1 + 2v_1 b_1}}{2v_1^2}\right).\end{aligned} \tag{4.30}$$

We note that a direct numerical integration of (4.17) is problematic since the inverse Laplace transform of $\Phi_{\mathrm{m-log}}(s|H_i)$ has discontinuities (reflected e.g. by the first and last terms in the sum on the right hand sides of (4.27), (4.29)

and (4.30)). However, the terms corresponding to the discontinuities can be easily inverted in closed form, and the remaining term without discontinuities can then be inverted numerically, e.g. using the Gauss–Chebyshev quadrature rules mentioned in section (4.3). Specifically, $\Phi_z(s|H_i)$ can be decomposed to

$$\Phi_z(s|H_i) = \Phi_{z_0}(s|H_i) + \sum_{p=1}^{P} C_{p,i} e^{-sJ_{p,i}}, \tag{4.31}$$

where P is the number of terms that contains discontinuities, $\Phi_{z_0}(s|H_i) = \Phi_z(s|H_i) - \sum_{p=1}^{P} C_{p,i} e^{-sJ_{p,i}}$ represents the term without discontinuities, and $C_{p,i}$, $J_{p,i}$ represent their respective coefficients defined earlier from the Laplace transform of the pdf of z_k. For (4.19), $P = 4$ and in the case where the WSN consist of K sensors, (4.17) can be expressed as,

$$\begin{aligned}\Phi_{\text{m-log}}(s|H_i) &= \left(\sum_{p=1}^{P} C_{p,i} e^{-sJ_{p,i}}\right)^K + \sum_{k=1}^{K} \binom{K}{k} \left(\Phi_{z_0}(s|H_i)\right)^k \\ &\quad \times \left(\sum_{p=1}^{P} C_{p,i} e^{-sJ_{p,i}}\right)^{K-k} = \Phi^d_{\text{m-log}}(s|H_i) + \Phi^{nd}_{\text{m-log}}(s|H_i).\end{aligned} \tag{4.32}$$

The term $\Phi^d_{\text{m-log}}(s|H_i)$ represents all the discontinuities in $\Phi_{\text{m-log}}(s|H_i)$ and can be written as

$$\left(\sum_{p=1}^{P} C_{p,i} e^{-sJ_{p,i}}\right)^K = \sum_{d_1,d_2,\ldots,d_P} \binom{K}{d_1, d_2, \ldots, d_P} C_{1,i} C_{2,i} \ldots C_{P,i} e^{-s(d_1 J_{1,i} + \cdots + d_P J_{P,i})}, \tag{4.33}$$

where the summation is taken over all sequences of integer d_p such that $\sum_{p=1}^{P} d_p = K$ and $\binom{K}{d_1,d_2,\ldots,d_P} \triangleq \frac{K!}{d_1! d_2! \ldots d_P!}$. The inverse Laplace transform of (4.33) can easily be obtained, and by combining (4.16), (4.19), (4.27), (4.32), and the corresponding expression (4.29), (4.30) for $N > 2$, the probabilities of false alarm

and detection can be exactly (approximately) computed for $N = 2$ ($N > 2$)

$$\begin{aligned} P_{\mathrm{x_i}} &= \sum_{d_1,d_2,\ldots,d_P} \binom{K}{d_1, d_2, \ldots, d_P} C_{1,i} C_{2,i} \ldots C_{P,i} \\ &\quad \times \bar{u}(-\gamma_0 - [d_1 J_{1,i} + \ldots + d_P J_{P,i}]) \\ &+ \frac{1}{2\pi j} \int_{c-j\infty}^{c+j\infty} \Phi^{nd}_{\mathrm{m-log}}(s|H_i) e^{-\gamma_0 s} \frac{\mathrm{d}s}{s}, \end{aligned} \tag{4.34}$$

where $\mathrm{x}_0 = f_0$ and $\mathrm{x}_1 = d_0$ denote the probability of false alarm and detection respectively and $\Phi^{nd}_{\mathrm{m-log}}(s|H_i)$ contains terms which can be inverted numerically.

Chapter 5

Numerical and Simulation Results

This chapter presents numerical and simulation results for the proposed optimal and suboptimal fusion rules. Binary hypothesis and non–binary hypothesis testing will be presented separately. For all results shown in this chapter, the middle symbol of the observation window is used for detection, i.e., $\nu_0 = N/2$, since this leads to the best performance.

5.1 Binary Hypothesis Testing

In this subsection, in order to confirm our simulation results with the analytical results from Chapter 4, we assume $M = 2$, i.i.d. Rayleigh fading channels, identical sensors, and $P(H_0) = P(H_1) = 1/2$. All curves labeled with "Theory" (for $N = 2$) and "Approximation" (for $N = 6$) were generated using the analytical methods discussed in Chapter 4, while the remaining curves were obtained by computer simulation.

5.1.1 Error Probability

In Figs. 5.1 and 5.2, we examine the error probability $P_e \triangleq P_{f_0}P(H_0) + (1 - P_{d_0})P(H_1)$ of the suboptimal MSD fusion rules considered vs. E_b/N_0 for $BT = 0.1$ and $BT = 0$, respectively. Here, E_b is the total received average energy per bit (from all sensors), and N_0 denotes the one–sided power spectral density of the underlying continuous–time noise process. The decision threshold $\gamma_0 = 0$ was used for all fusion rules and $K = 8$, $P_d = 0.8$, and $P_f = 0.01$. In addition to the suboptimal MSD fusion rules, Figs. 5.1 and 5.2 also contain the two performance upper bounds introduced in Section 4.1. Furthermore, Fig. 5.1 also includes the performance of the optimal fusion rule for $N = 2$ (the optimal fusion rule is computationally not feasible for $N = 6$) and the error probability of the coherent max–log fusion rule for DPSK, whereas Fig. 5.2 shows the performance of the coherent versions of all three suboptimal MSD fusion rules. We note that these coherent fusion rules require perfect knowledge of the fading channel gains. Figs. 5.1 and 5.2 show that while the ILS fusion rule has the best performance for very low E_b/N_0, where transmission errors significantly affects the overall performance, the CV and the max–log fusion rules yield a superior performance for medium–to–high E_b/N_0. A comparison of Figs. 5.1 and 5.2 reveals that increasing the observation window size from $N = 2$ to $N = 6$ is more beneficial for fast fading ($BT = 0.1$) than for static fading ($BT = 0$). In the latter case, the performance gap between coherent detection and the respective MSD fusion rules is relatively small even for $N = 2$. In contrast, for $BT = 0.1$ and $N = 2$ the performance of both the CV and the max–log fusion rules is limited by the high error floor caused by the fast fading. This high error floor is also seen under the optimal fusion rule, which yields a negligible performance gain compared to the max–log fusion rule for $N = 2$. For $N = 6$ this error floor is mitigated and both the CV and the max–log fusion rules approach Bound II for high E_b/N_0, i.e., performance is limited by local sensor

decision errors in this case and not by transmission errors. For the ILS fusion rule increasing the observation window to $N > 2$ is not beneficial and it even leads to a loss in performance for high E_b/N_0 for $BT = 0$. This somewhat surprising behavior is caused by the local decision errors at the sensors, which were ignored for derivation of the ILS fusion rule. For $N = 2$, theoretical and simulation results in Figs. 5.1 and 5.2 match perfectly, confirming the analysis in Section 4. As expected from the discussions in Section 4, for $N = 6$, there is a good agreement between theoretical and simulation results for the CV and the max–log fusion rules at high E_b/N_0 ratios, cf. Fig. 5.1 (for clarity of presentation the analytical curves for $N = 6$ have been omitted in Fig. 5.2). At low E_b/N_0 ratios, the analytical results overestimate the actual P_e since the assumptions leading to the analytical result for $N > 2$ are less justified.

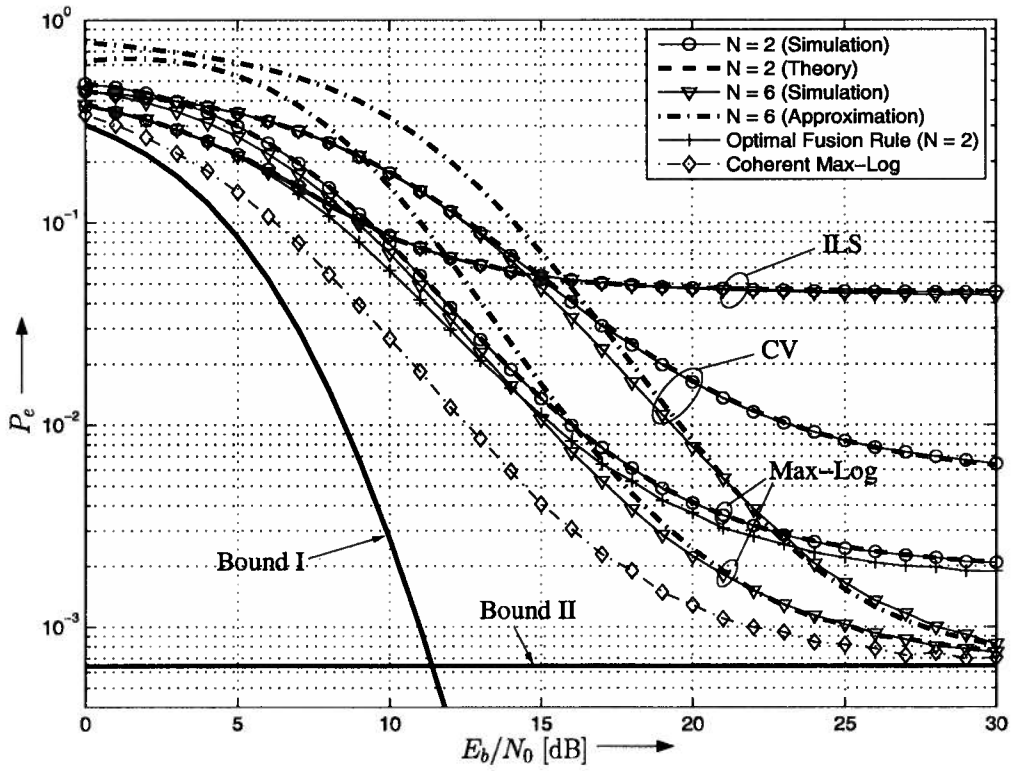

Figure 5.1: Probability of error P_e vs. E_b/N_0 for decision threshold $\gamma_0 = 0$. $K = 8$, $M = 2$, $P_d = 0.8$, $P_f = 0.01$, $BT = 0.1$, and i.i.d. Rayleigh fading.

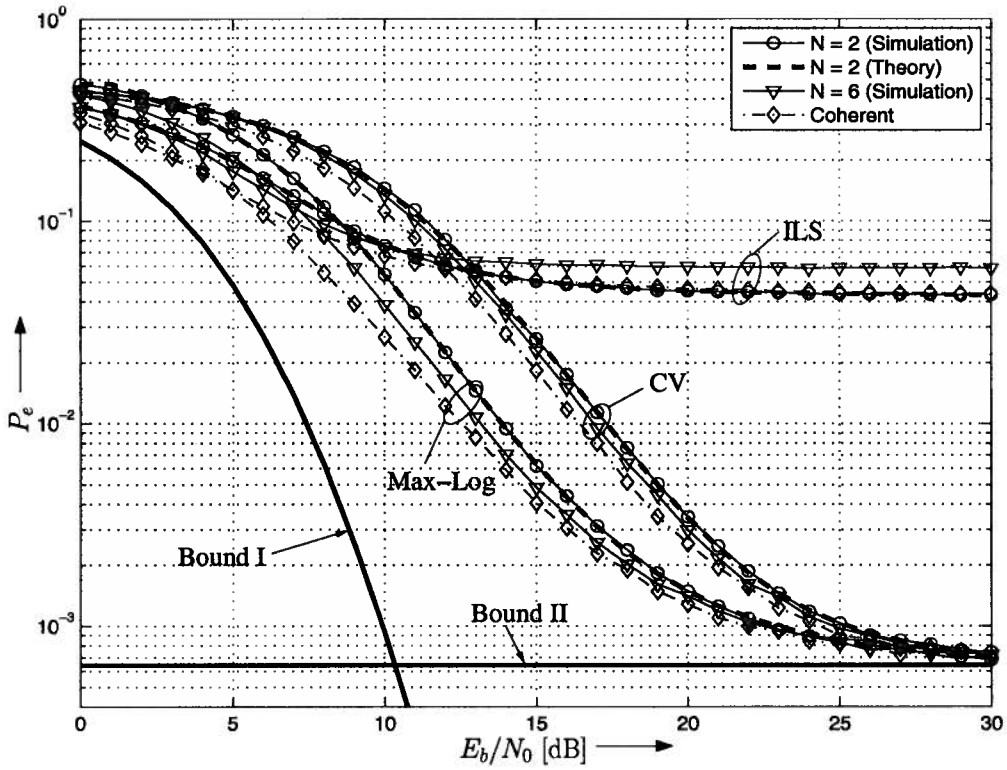

Figure 5.2: Probability of error P_e vs. E_b/N_0 for decision threshold $\gamma_0 = 0$. $K = 8$, $M = 2$, $P_d = 0.8$, $P_f = 0.01$, $BT = 0$, and i.i.d. Rayleigh fading.

5.1.2 Constant False Alarm Probability

Fig. 5.3 shows P_{d_0} as a function of E_b/N_0 for a fixed probability of false alarm of $P_{f_0} = 0.001$, which is achieved by adjusting decision threshold γ_0 accordingly. Furthermore, $K = 8$, $P_d = 0.7$, $P_f = 0.05$, and $BT = 0.1$. In Fig. 5.3, the max–log fusion rule yields a superior performance compared to the other suboptimal MSD fusion rules but the CV and ILS fusion rules approach the max–log performance for high and low E_b/N_0, respectively. For $N = 6$, both the max–log and the CV fusion rules approach Bound II for high enough E_b/N_0, whereas for $N = 2$, these fusion rules as well as the optimal fusion rule are limited by transmission errors caused by fast fading. In contrast, the ILS fusion rule achieves a better performance for $N = 2$ than for $N = 6$. Fig. 5.3

shows again a good agreement between analytical and simulation results.

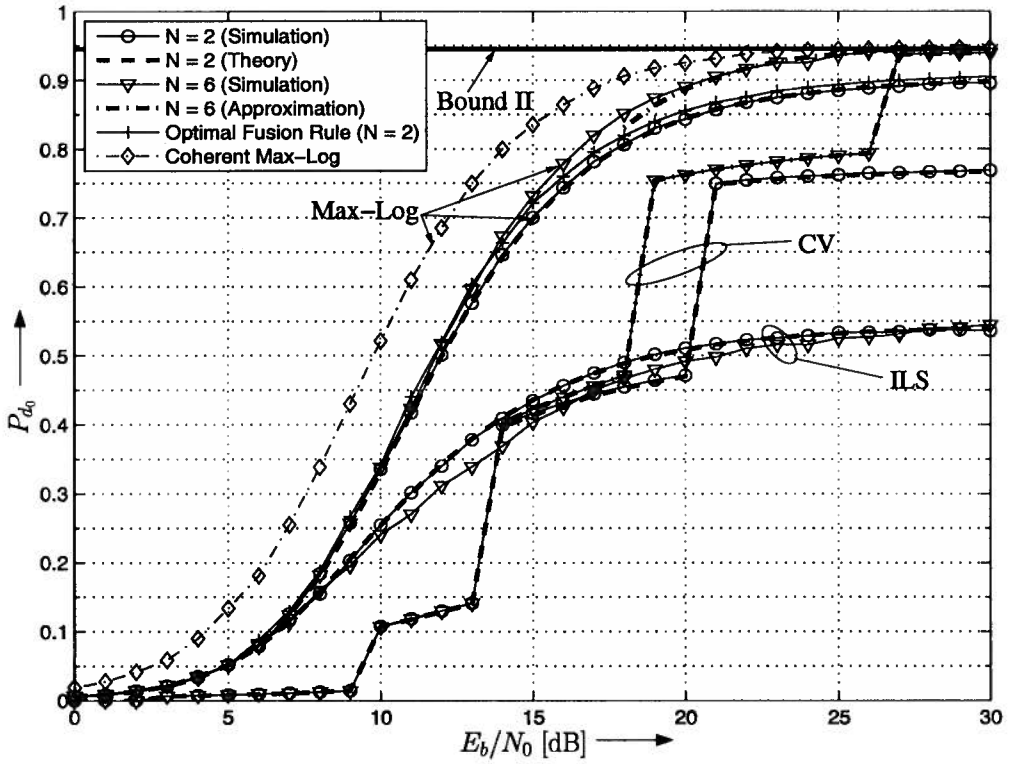

Figure 5.3: Probability of detection P_{d_0} vs. E_b/N_0 for a probability of false alarm of $P_{f_0} = 0.001$. $K = 8$, $M = 2$, $P_d = 0.7$, $P_f = 0.05$, $BT = 0.1$, and i.i.d. Rayleigh fading.

5.1.3 Receiver Operating Characteristic

In Fig. 5.4, we show the ROC for the considered MSD fusion rules and the coherent max–log fusion rule for DPSK. $K = 8$, $P_d = 0.7$, $P_f = 0.05$, $BT = 0.1$, and $E_b/N_0 = 20$ dB. Fig. 5.4 shows the superiority of the max–log fusion rule especially if low probabilities of false alarm are desired. Increasing N from two to six yields significant gains for both the max–log and the CV fusion rules. In fact, the max–log fusion rule with $N = 6$ bridges half of the performance gap between the coherent max–log fusion rule and the MSD max–log fusion rule with $N = 2$. On the other hand, for $N = 2$ the optimal fusion rule performs only slightly better than the max–log fusion rule.

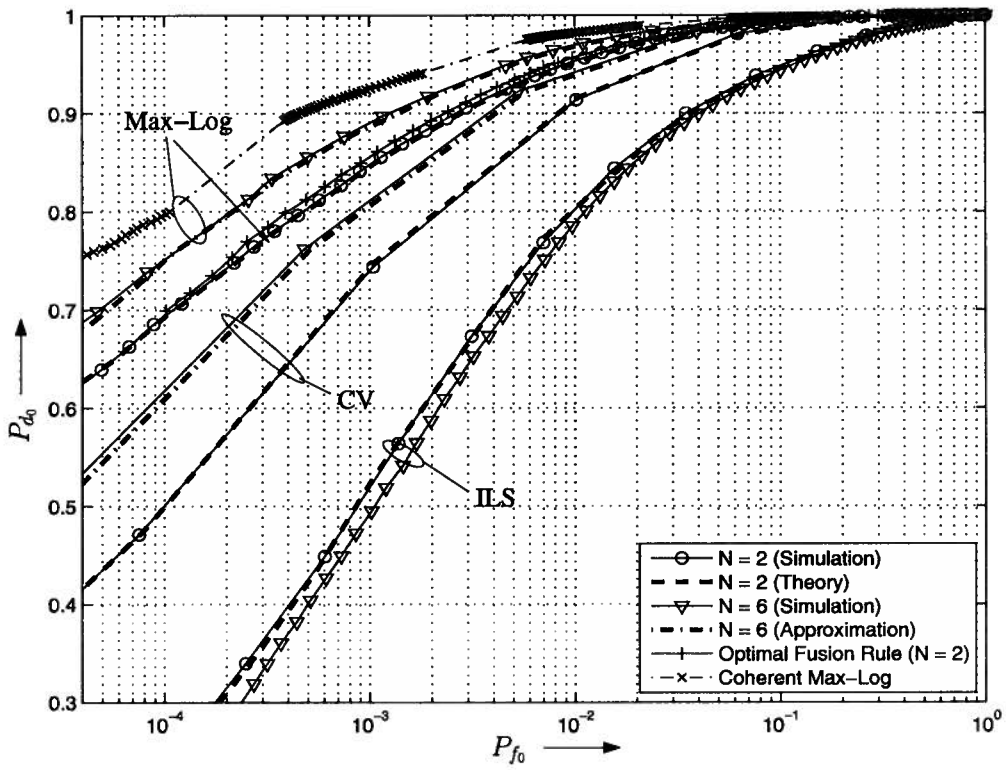

Figure 5.4: Probability of detection P_{d_0} vs. probability of false alarm P_{f_0}. $K = 8$, $M = 2$, $P_d = 0.7$, $P_f = 0.05$, $BT = 0.1$, $E_b/N_0 = 20$ dB, and i.i.d. Rayleigh fading.

5.1.4 Effects of Sensor Network Size

Figs. 5.5 and 5.6 show the impact of the number of sensors on P_e and P_{d_0}, respectively, for $P_d = 0.7$, $P_f = 0.05$, $BT = 0.1$, and $E_b/N_0 = 20$ dB. For Fig. 5.5, we optimized the decision threshold γ_0 for minimization of P_e for each fusion rule and each considered K. For Fig. 5.6, γ_0 was chosen so as to guarantee $P_{f_0} = 0.001$. Figs. 5.5 and 5.6 indicate that the max–log fusion rule benefits more from an increasing number of sensors than the ILS and the CV fusion rules. In particular, the CV fusion rule shows a saturation effect for large K in Fig. 5.5. This is due to the fact that since the total E_b/N_0 of all sensors is fixed, the channel SNR per sensor decreases as K increases. Therefore, the assumption of a perfect transmission channel, which was implicitly made for derivation of the CV fusion rule, becomes less justified as K increases leading

to a loss in performance.

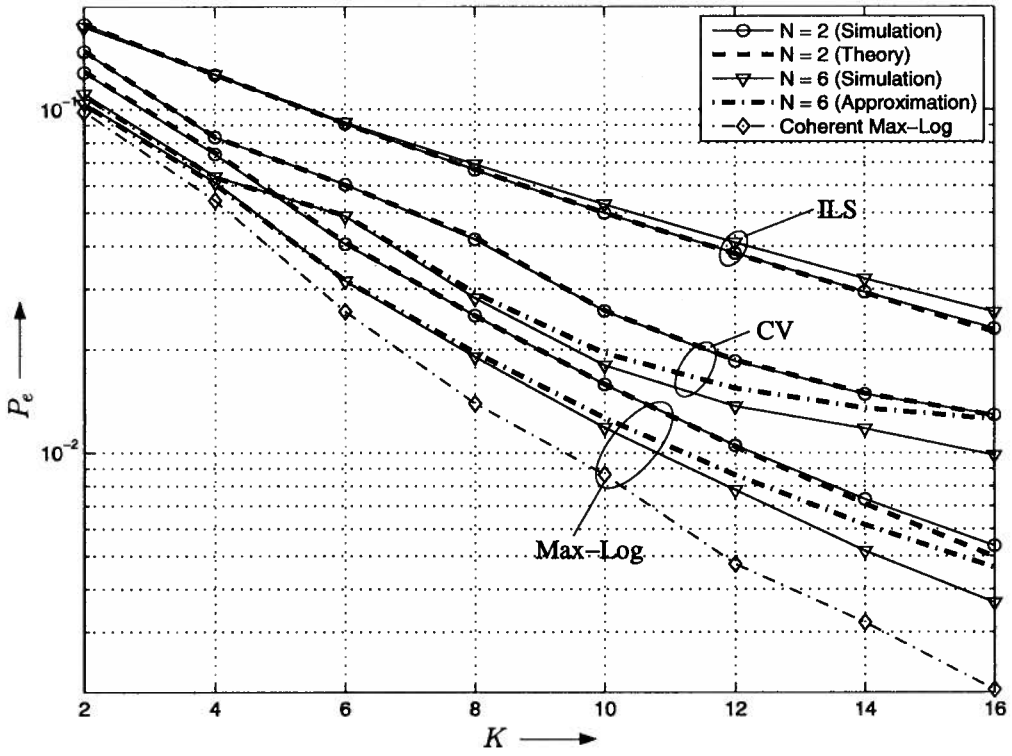

Figure 5.5: Probability of error P_e vs. number of sensors K. $M = 2$, $P_d = 0.7$, $P_f = 0.05$, $BT = 0.1$, $E_b/N_0 = 20$ dB, and i.i.d. Rayleigh fading.

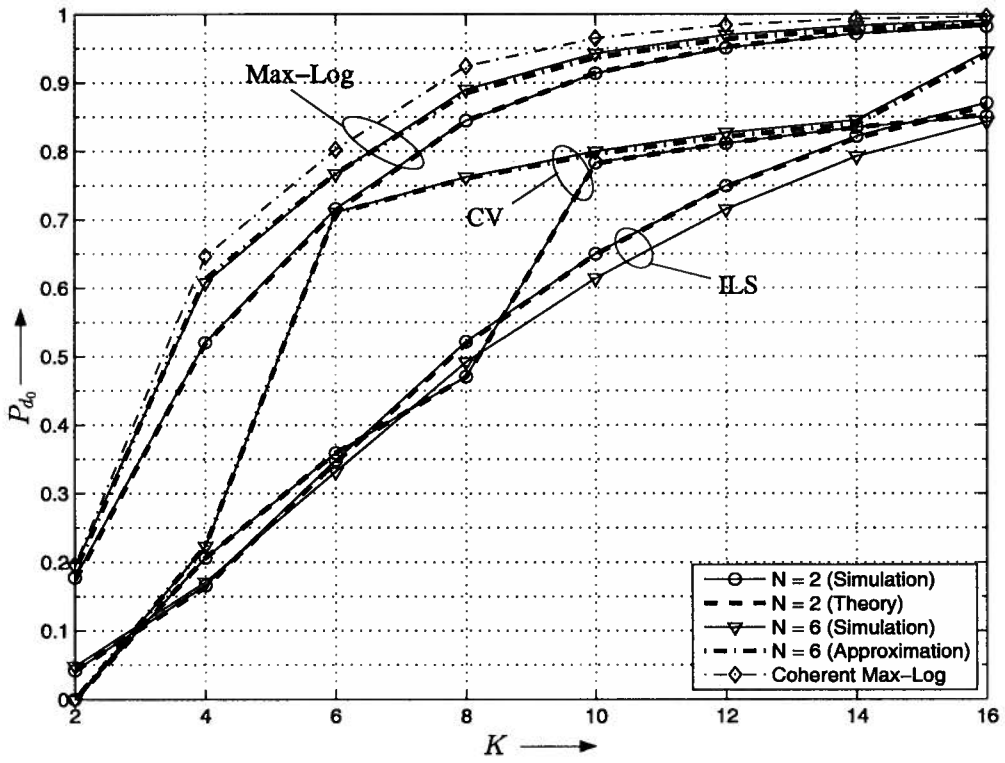

Figure 5.6: Probability of detection P_{d_0} vs. number of sensors K for a probability of false alarm of $P_{f_0} = 0.001$. $M = 2$, $P_d = 0.7$, $P_f = 0.05$, $BT = 0.1$, $E_b/N_0 = 20$ dB, and i.i.d. Rayleigh fading.

5.2 Multiple Hypothesis Testing

For the multiple hypothesis testing case, we assume that the local sensor observations are given by $x_k[n] = u_k[n] + \tilde{n}_k[n]$, $k \in \mathcal{K}$, where $u_k[n] \in \{-(M-1), -(M-3), \ldots, M-1\}$ and $\tilde{n}_k[n]$ is real AWGN. Throughout this subsection, we assume identical sensors, $P(H_i) = 1/M$, $i \in \mathcal{M}$, and $M = 4$. The sensor performance indices $P_k(a_k[n] = w_j|H_i)$, $i \in \mathcal{M}$, $j \in \mathcal{M}$, $k \in \mathcal{K}$, depend on the sensor SNR, $\text{SNR}_s \triangleq \mathcal{E}\{|u_k[n]|^2\}/\mathcal{E}\{|\tilde{n}_k[n]|^2\}$.

5.2.1 Error Probability

In Fig. 5.7, we show the probability of missed detection $P_m \triangleq \sum_{i=0}^{M-1} \sum_{\hat{i} \neq i} P(H_{\hat{i}}|H_i)P(H_i)$ vs. E_b/N_0 for the proposed suboptimal MSD fusion rules and the corresponding coherent fusion rules. Again E_b is defined as the total received average energy per bit (from all sensors) and $K = 8$, $BT = 0$, $\text{SNR}_s = -3$ dB and i.i.d. Rayleigh fading channels. Similar to the binary hypothesis error probability curves in static fading ($BT = 0$), the ILS fusion rule performs best for very low E_b/N_0, the CV and max–log fusion rules perform well in the middle–to–high E_b/N_0 region. With $N = 2$ the performance is already relatively close to coherent detection and when the observation window is increased to $N = 6$, only a minor increase in performance is observed. Again it is observed that increasing the observation window size actually increases the missed detection if the ILS fusion rule is used. On the other hand, in Fig. 5.8 where the proposed MSD fusion rules are shown for $N = 2, 4, 6$ when $BT = 0.1$, it is easily seen that increasing the observation window is beneficial. As the observation window is increased to $N = 6$, the CV and max–log fusion rules are able to reduce the inherent error floor caused by fast fading and its error rate approaches the performance limit due to decision errors made by the local sensors. From Figs. 5.7 and 5.8, we conclude that, for ILS fusion, increasing the observation size is not recommended. When the observation size is increased to $N = 6$, the performance degrades in static channels while in non–static channels only minimal improvement is observed while the complexity of the fusion rule increases dramatically.

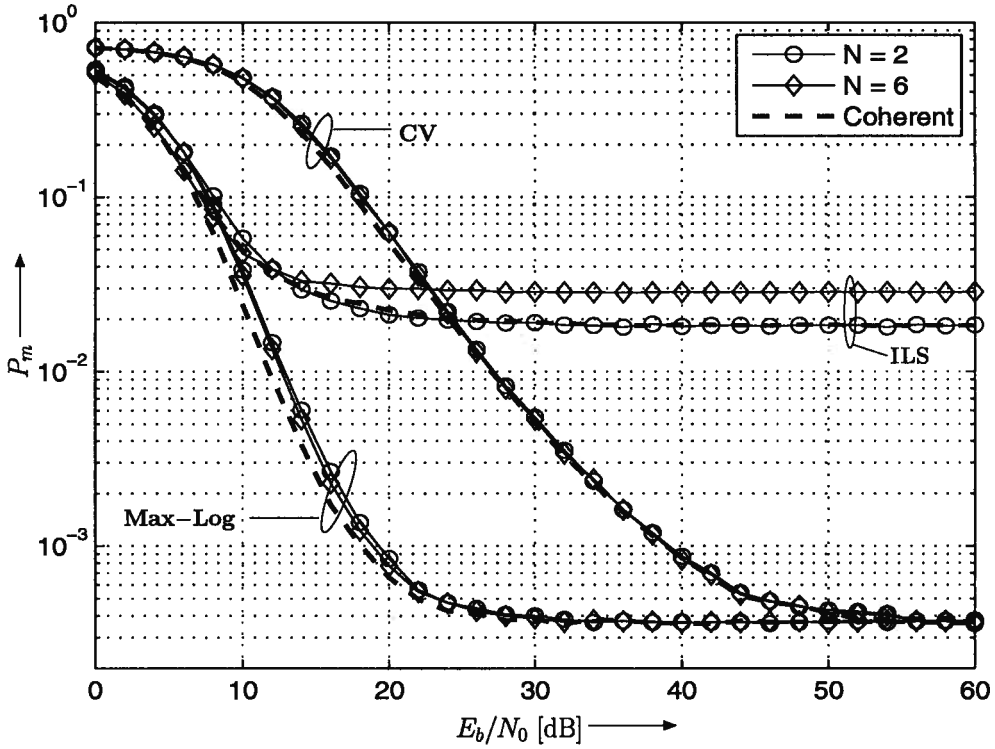

Figure 5.7: Probability of missed detection P_m vs. E_b/N_0. $M = 4, BT = 0$, $\text{SNR}_s = -3$ dB, and i.i.d. Rayleigh fading.

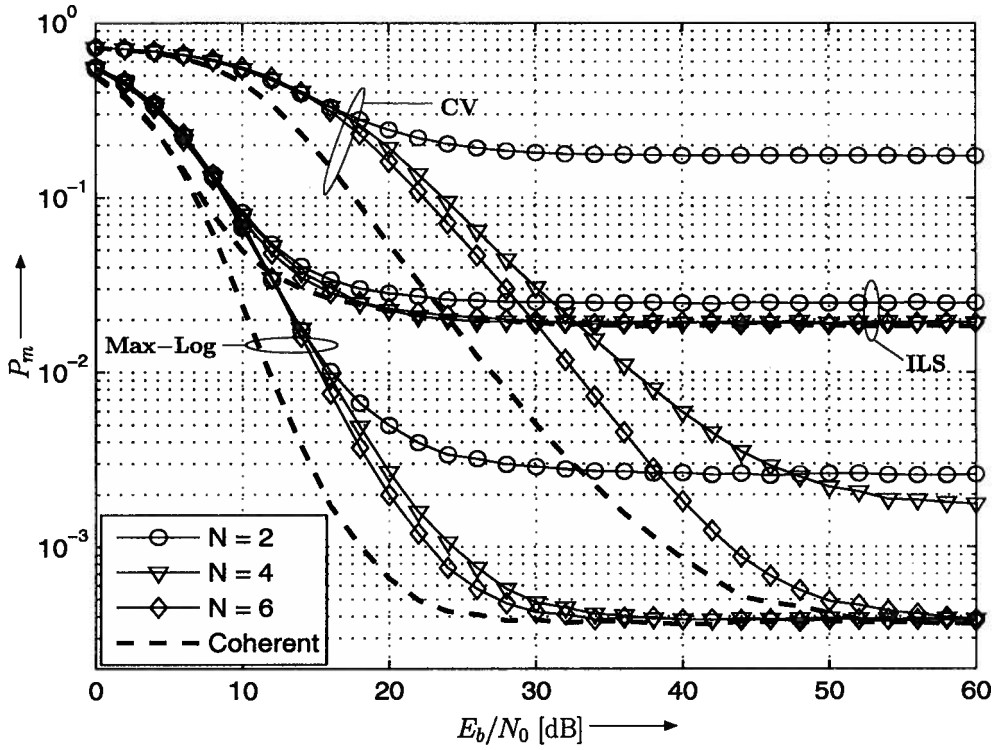

Figure 5.8: Probability of missed detection P_m vs. E_b/N_0. $M = 4, BT = 0.1$, $\text{SNR}_s = -3$ dB, and i.i.d. Rayleigh fading.

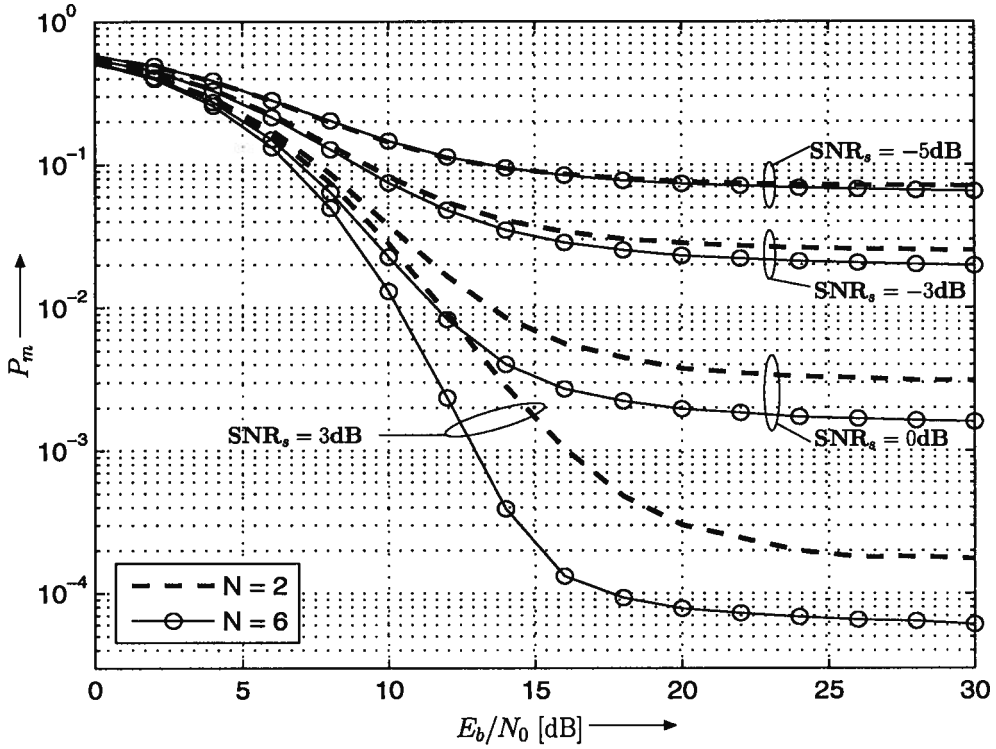

Figure 5.9: Probability of missed detection P_m vs. E_b/N_0 for ILS fusion rule. $M = 4, BT = 0.1,$, i.i.d. Rayleigh fading, and SNR$_s$ = −5 dB, −3 dB, 0 dB, 3 dB.

Fig. 5.9 shows miss detection vs. E_b/N_0 using ILS fusion for various SNR$_s$. In the derivation of the ILS rule in Section 3.3, as SNR$_s$ increases, i.e., with more reliable sensor decisions, the benefit of increasing the observation window size to $N = 6$ becomes more apparent. Therefore, if the sensors can be made more reliable, e.g., by time averaging the observation at the sensors to improve SNR$_s$, increasing the observation size can be considered for ILS fusion, otherwise a more conservative approach should be taken by using $N = 2$.

5.2.2 Independent, Non–identically Distributed Fading

Fig. 5.10 shows the probability of missed detection vs. E_b/N_0 for the proposed suboptimal MSD fusion rules and the corresponding coherent fusion rules with

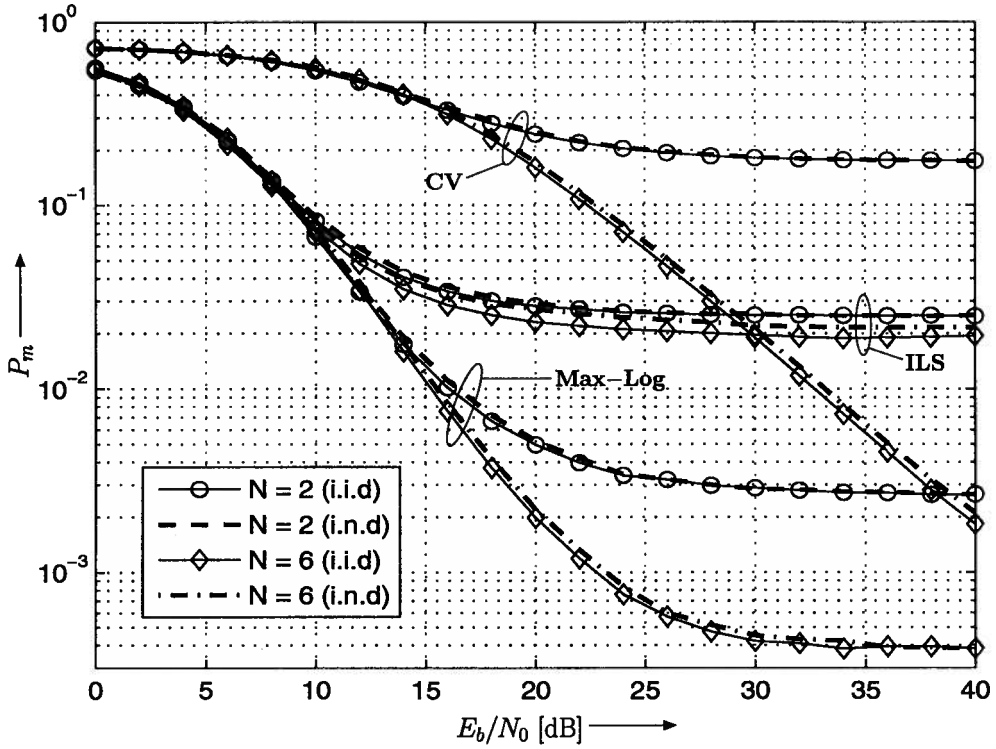

Figure 5.10: Probability of missed detection P_m vs. E_b/N_0. $M = 4, BT = 0.1$, $\text{SNR}_s = -3$ dB, and i.n.d. Rayleigh fading.

the following parameters: $K = 8$, $BT = 0.1$, $\text{SNR}_s = -3$ dB, and channel SNR of sensors $k \in \{1, 2, 3, 4\}$ was 3 dB higher than that of the remaining four sensors, i.e., the fading was i.n.d. As a reference, the performance in i.i.d. fading channels are also shown. Interestingly both the CV and max–log fusion rules show little performance degradation for i.n.d. fading for both considered observation window sizes. Although the ILS fusion rule also shows little degradation in missed detection performance, the negative effect of i.n.d. fading is more apparent as the observation window size increases. Since ILS fusion is limited by the fact that it is ignoring sensor reliability, weighting the sensors with more reliable channels effectively weights certain sensors more even though all sensors are identical and unreliable. In the extreme case if the sensors are not identical, and the SNR_s are lower at the sensors with higher

channel SNR, the missed detection performance can degrade dramatically.

5.2.3 Effects of Sensor Reliability

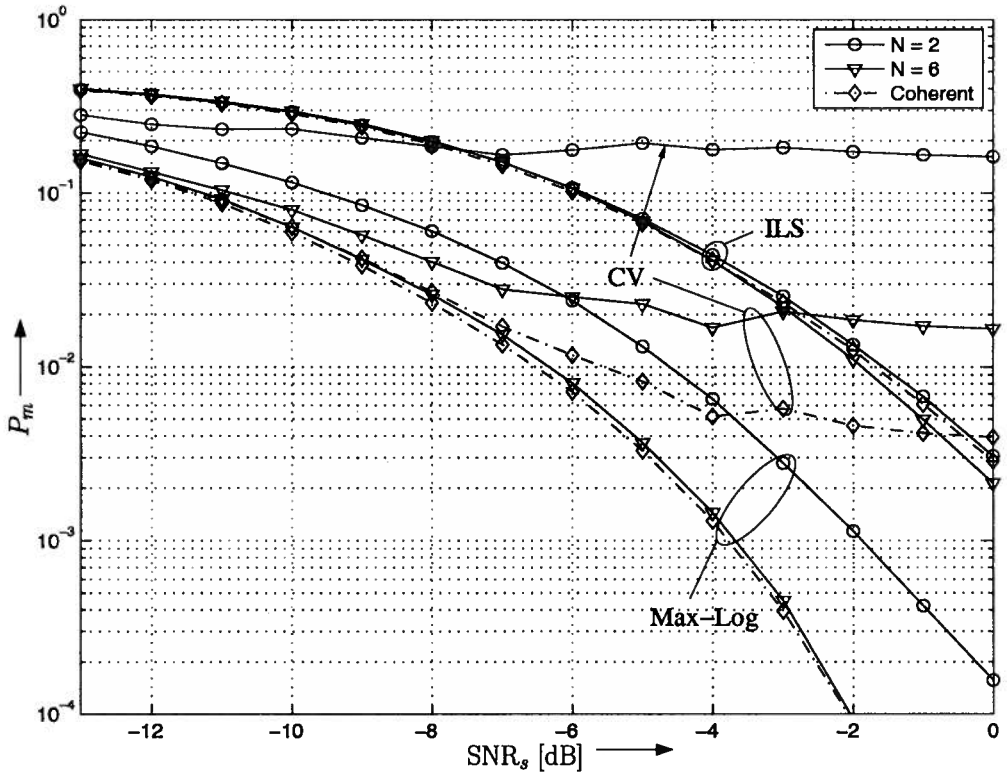

Figure 5.11: Probability of missed detection P_m vs. SNR$_s$. $M = 4, BT = 0.1, E_b/N_0 = 30$ dB, and i.n.d. Rayleigh fading.

In Fig. 5.11, we show the probability of missed detection as a function of SNR$_s$ for the proposed suboptimal MSD fusion rules and the corresponding coherent fusion rules with the following parameters: $K = 8$, $BT = 0.1$, $E_b/N_0 = 30$ dB, and i.n.d. channel (four channels have a 3 dB higher SNR than the remaining four channels). For low sensor SNR, the CV fusion rule achieves a similar performance as the max–log fusion rule since the overall performance is mainly affected by the unreliable sensors. However, the CV fusion rule is not able to fully exploit the increasing reliability of the sensors when the sensor SNR improves and is ultimately limited by an error floor caused by

transmission errors which are not optimally taken into account in the CV fusion rule. With highly reliable sensors, the CV fusion rule is even outperformed by the ILS fusion rule, whose performance steadily improves with increasing sensor SNR. This is probably because the assumption on which the ILS fusion rule is based, namely error–free sensors, has become more and more justified at high sensor SNR. Nevertheless, the max–log fusion rule yields the best performance among all considered MSD fusion rules, and closely approaches the performance of the coherent max–log fusion rule with $N = 6$.

5.2.4 Estimation Error of Channel Correlation Matrix

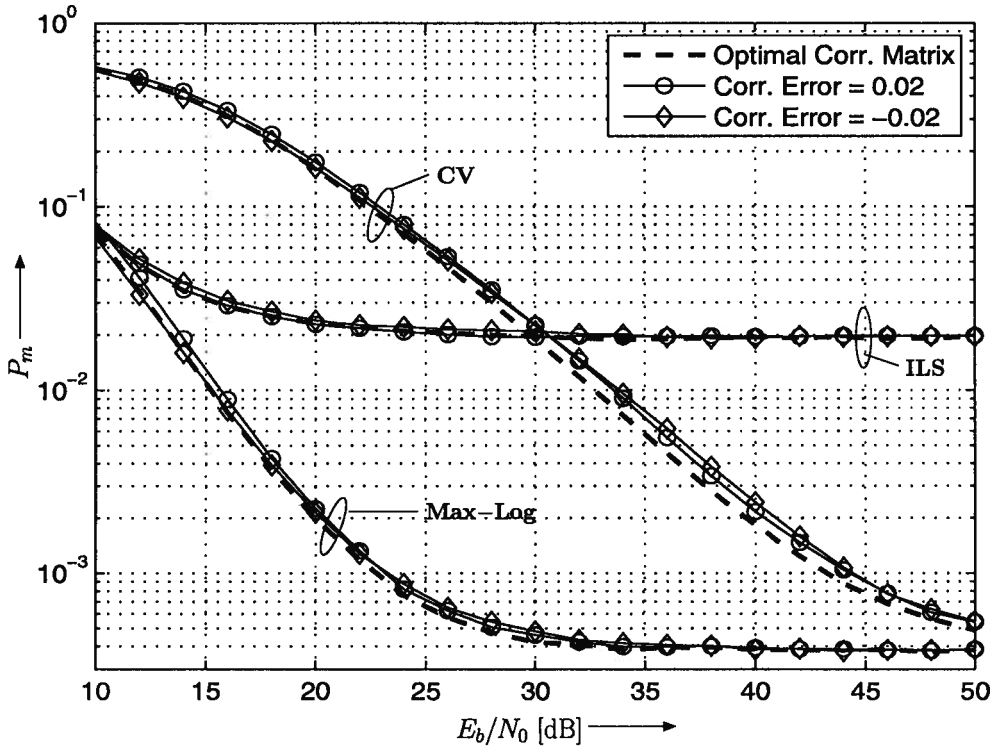

Figure 5.12: Probability of missed detection P_m vs. E_b/N_0 with channel correlation error. $M = 4, BT = 0.1$, $\text{SNR}_s = -3$ dB, and i.i.d. Rayleigh fading.

In Fig. 5.12, we show the probability of missed detection as a function

of E_b/N_0 for the proposed suboptimal MSD fusion rules and consider the effect of estimation error on the channel correlation matrix, $\boldsymbol{R}_{hh}$, i.e., $\varphi_{hh}[\lambda] = \sigma^2 J_0(2\pi(B+\Delta_e)T\lambda)$. $K = 8$, $BT = 0.1$, $\mathrm{SNR}_s = -3$ dB, and a correlation error $\Delta_e = \pm 0.02$, i.e., the estimated fading bandwidth is $BT = 0.08$ and $BT = 0.12$, respectively. A small deviation is shown for all three fusion rules relative to the curve when the optimal matrix is used. For both max–log and ILS fusion rules a minimal increase in missed detection is observed. For low E_b/N_0, the max–log fusion rule exhibits the largest missed detection difference since the channel errors are the limiting factor at this operating range. When ILS fusion is performed, the performance degradation when using a non–optimal channel matrix for detection has little impact on fusion performance. On the other hand, for the CV fusion rule at medium–to–high SNRs exhibits a larger performance degradation of approximately 1dB. This is due to the assumption of perfect channel conditions for the CV fusion rule. Nevertheless, all three proposed fusion rules show a high tolerance to estimation errors in the channel correlation matrix.

5.3 Computational Complexity

In Fig. 5.13, we compare the complexity of the considered MSD fusion rules for $N = 6$ as a function of E_b/N_0 with the following parameters: $K = 8$, $M = 4$, i.i.d. Rayleigh fading, and $\mathrm{SNR}_s = -3$ dB. The complexity is measured in terms of the (average) number of real multiplications required per decision. The dashed lines in Fig. 5.13 denote the number of multiplications required by the respective sphere decoders to find the first vector $\hat{\boldsymbol{s}}$ and constitute the lower bounds for the actual complexity. Note that the lower bounds for the CV and ILS fusion rules practically coincide for the considered example. Fig. 5.13 shows that the CV fusion rule closely approaches the corresponding lower bound. In contrast, for the ILS and max–log fusion rules there is always

a considerable gap between the actual complexity and the lower bound even at high E_b/N_0. For the ILS fusion rule, this gap is due to erroneous sensor decisions as can be observed from the comparison with the (hypothetical) case of ideal local sensor decisions. For the max–log fusion rule the gap is due to the fact that the sphere decoder does not only have to find the ML vector as for the CV and ILS fusion rules but also has to perform a constrained search over all $\boldsymbol{a}_k$ with $a(\nu_0) = w_j$, $j \in \mathcal{M}$, cf. Section 3.4. Nevertheless, all three suboptimal fusion rules have a significantly lower complexity than the optimal fusion rule.

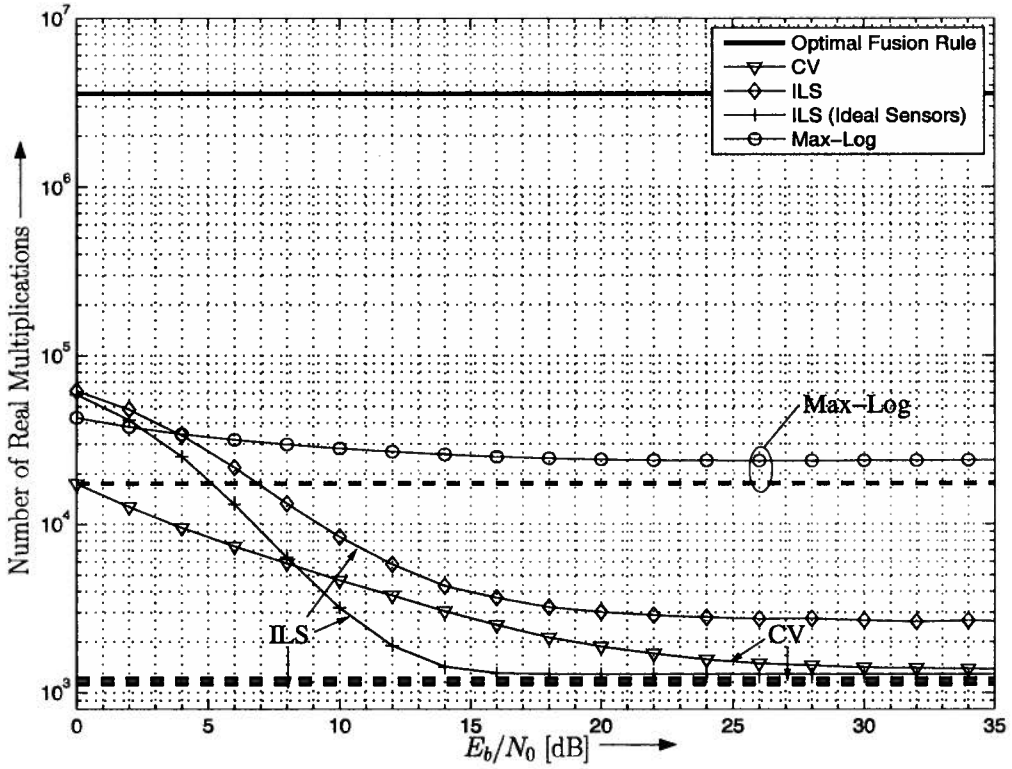

Figure 5.13: Number of real multiplications per decision vs. E_b/N_0. $M = 4$, $\mathrm{SNR}_s = -3$ dB, and i.i.d. Rayleigh fading.

Chapter 6

Conclusions and Future Work

6.1 Research Contributions

In this thesis, we have considered the distributed multiple hypothesis testing problem for mobile wireless sensor networks where sensors employ DPSK to cope with time–variant fading. We have shown that since differential modulation introduces memory, it is advantageous to consider fusion rules that base their decisions on an observation window of multiple symbol intervals. Specifically, we have derived the optimal MSD fusion rule (where complexity increases exponentially with an increase in the number of sensors and the observation window size), and three suboptimal MSD fusion rules, where complexity is linear in the number of sensors and, at high SNRs, polynomial in the observation window size. For binary hypothesis testing, performance bounds for the optimal fusion rule have been derived, and for the suboptimal fusion rules, exact or approximate expressions for the probabilities of false alarm and detection have been provided. Our simulation and analytical results show that for high and low channel SNR respectively, the performance of the CV and ILS fusion rules approach that of the optimal fusion rule. The proposed max–log fusion rule achieves a close–to–optimal performance over the entire SNR range

but has a higher complexity than the CV and ILS fusion rules.

6.2 Future Work

There are several possible extensions of the work presented in this thesis and these are listed below.

The first is to consider necessary modifications to the proposed fusion rules if they are adapted for different WSN topologies as discussed in Section 1.1, thereby giving a more generalized solution to the WSN decision fusion problem.

The second is to consider whether our assumption of Rayleigh fading can be extended to other fading models. Rician fading should be investigated since there are viable scenarios where the FC and the sensors have direct line–of–sight. In particular, new sub-optimal fusion rules can be investigated since in Rician fading MSDSD is not optimal.

A third area for futher development is error correction coding in WSN. The possibility of using error correcting codes to improve detection rates specifically at lower channel SNR can dramatically improve the CV fusion rule performance. Moreover, coding can provide a means of reducing the problems related to faulty sensors that repeatedly send erroneous decisions due to sensor malfunctions.

Bibliography

[1] R. Viswanathan and P. K. Varshney. Distributed detection with multiple sensors: Part I–Fundamentals. *Proceedings of the IEEE*, 85(1):54–63, January 1997.

[2] L. Song and D. Hatzinakos. Architecture of wireless sensor networks with mobile sinks: sparsely deployed sensors. *IEEE Trans. Veh. Technol.*, 56:1826–1836, July 2007.

[3] A. Durresi, M. Durresi, and L. Barolli. Secure mobile communications for battlefields. In *Proceedings of the International Conference Complex, Intelligent and Software Intensive Systems*, pages 205–210, March 2008.

[4] S. M. Mishra, A. Sahai, and R. W. Brodersen. Cooperative sensing among cognitive radios. In *Proceedings of the IEEE International Conference on Communications (ICC'05)*, pages 11–15, Turkey, June 2006.

[5] Z. Quan, S. Cui, A.H. Sayed, and H.V. Poor. Wideband spectrum sensing in cognitive radio networks. In *Proceedings of the IEEE International Conference on Communications (ICC'08)*, pages 901–906, Turkey, May 2008.

[6] A. D'Costa, V. Ramachandran, and A. M. Sayeed. Distributed classification of gaussian space-time sources in wireless sensor networks. *IEEE Trans. Commun.*, 22:1026–1036, August 2004.

[7] J. H. Kotecha, V. Ramachandran, and A. M. Sayeed. Distributed multi-target classification in wireless sensor networks. *IEEE Trans. Commun.*, 23:703–713, April 2005.

[8] R.W. Brodersen, A. Wolisz, D. Cabric, S.M. Mishra, and D. Willkomm. A cognitive radio approach for usage of virtual unlicensed spectrum. In *Berkeley, CA:Univ. California Berkeley Whitepaper*, July 2004.

[9] J. Unnikrishnan and V.V. Veeravalli. Cooperative sensing for primary detection in cognitive radio. *IEEE Trans. Signal Processing*, 2:18–27, February 2008.

[10] Z. Quan, S. Cui, and A.H. Sayed. Optimal linear cooperation for spectrum sensing in cognitive radio networks. *IEEE Trans. Signal Processing*, 2(1):28–40, February 2008.

[11] P. K. Varshney. *Distributed Detection and Data Fusion.* Springer–Verlag, New York, 1997.

[12] R. Tenney and N. Sandell, Jr.. Detection with distributed sensors. *IEEE Trans. on Aerospace and Electronics Systems*, 17(4):501–510, July 1981.

[13] J. N. Tsitsiklis. Decentralized detection. *Advances in Statistical Signal Processing,* JAI Press Inc., 2:297–344, 1993.

[14] R. S. Blum, S. A. Kassam, and H. V. Poor. Distributed detection with multiple sensors: Part II–Advanced topics. *Proceedings of the IEEE*, 85(1):64–79, January 1997.

[15] B. Chen, R. Jiang, T. Kasetkasem, and P. K. Varshney. Channel aware decision fusion in wireless sensor networks. *IEEE Trans. Signal Processing*, 52:3454–3458, December 2004.

[16] R. Niu, B. Chen, and P. K. Varshney. Fusion of decisions transmitted over Rayleigh fading channels in wireless sensor networks. *IEEE Trans. Signal Processing*, 54:1018–1027, March 2006.

[17] R. Jiang and B. Chen. Fusion of censored decisions in wireless sensor networks. *IEEE Trans. Wireless Commun.*, 4:2668–2673, Nov. 2005.

[18] J.-F. Chamberland and V. Veeravalli. The impact of fading on decentralized detection in power constrained wireless sensor networks. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 17–21, May 2004.

[19] G. Mergen, V. Naware, and L. Tong. Asymptotic detection performance of type-based multiple access over multiaccess fading channels. *IEEE Trans. Signal Processing*, 55:1081–1092, March 2007.

[20] T. Wimalajeewa and S. Jayaweera. Optimal power scheduling for correlated data fusion in wireless sensor networks via constraint PSO. *IEEE Trans. Wireless Commun.*, 7:3608–3618, September 2008.

[21] V. Kanchumarthy, R. Viswanathan, and M. Madishetty. Impact of channel errors on decentralized detection performance of wireless sensor networks: a study of binary modulations, Rayleigh-fading and nonfading channels, and fusion-combiners. *IEEE Trans. Signal Processing*, 56:1761–1769, May 2008.

[22] J.G. Proakis. *Digital Communications.* McGraw–Hill, New York, fourth edition, 2000.

[23] D. Divsalar and M. K. Simon. Multiple-symbol differential detection of MPSK. *IEEE Trans. Commun.*, 38:300–308, March 1990.

[24] D. Fung and P. Ho. Error performance of multiple-symbol differential detection of PSK signals transmitted over correlated Rayleigh fading channels. *IEEE Trans. Commun.*, 40:1566–1569, October 1992.

[25] D. Divsalar and M. K. Simon. Maximum-likelihood differential detection of uncoded and trellis coded amplitude phase modulation over AWGN and fading channels–Metrics and performance. *IEEE Trans. Commun.*, 42:76–89, January 1994.

[26] V. Pauli, R. Schober, and L. Lampe. A unified performance analysis framework for differential detection in MIMO Rayleigh fading channels. *IEEE Trans. Commun.*, 56:1972–1981, November 2008.

[27] L. Lampe, R. Schober, V. Pauli, and C. Windpassinger. Multiple-symbol differential sphere decoding. *IEEE Trans. Commun.*, 53:1981–1985, December 2005.

[28] M.O. Damen, H.E. Gamal, and G. Caire. On maximum-likelihood detection and the search for the closest lattice point. *IEEE Trans. Inform. Theory*, 49(10):2389–2402, October 2003.

[29] A. Burg, M. Borgmann, M. Wenk, C. Studer, and H. Bolcskei. Advanced receiver algorithms for MIMO wireless communications. In *Proceedings on Design, Automation and Test in Europe*, volume 1, pages 593–598, March 2006.

[30] Z. Guo and P. Nilsson. Algorithm and implementation of the k–best sphere decoding for MIMO detection. *IEEE Trans. Commun.*, 24(3):491–503, March 2006.

[31] R. Schober, W. H. Gerstacker, and J. B. Huber. Decision–feedback differential detection of MDPSK for flat Rayleigh fading channels. *IEEE Trans. Commun.*, 47:1025–1035, July 1999.

[32] R. Schober and W. H. Gerstacker. Decision-feedback differential detection based on linear prediction for MDPSK signals transmitted over Ricean fading channels. *IEEE J. Select. Areas Commun.*, 18:391–402, March 2000.

[33] P. Hoeher, P. Robertson, and E. Villebrun. A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain. In *Proceedings of the IEEE International Conference on Communications (ICC)*, pages 1009–1013, Seattle, June 1995.

[34] J. Jalden and B. Ottersten. Parallel implementation of a soft output sphere decoder. *Proc. of Asilomar Conference on Signals, Systems and Computers*, pages 581–585, November 2005.

[35] E. Biglieri, G. Caire, G. Taricco, and J. Ventura-Traveset. Computing error probabilities over fading channels: A unified approach. *European Transactions on Telecommunications*, 9:15–25, January-Feburary 1998.

[36] H. David and H. Nagaraja. *Order Statistics.* Wiley, 2003.