Implementations of Signal-Processing Algorithms for OFDM Systems

Guest Editors: Ming-Der Shieh, Yin-Tsong Hwang, Hanho Lee, Chirn Chye Boon, and Zhiyuan Yan
Implementations of Signal-Processing Algorithms for OFDM Systems
Implementations of Signal-Processing Algorithms for OFDM Systems

Guest Editors: Ming-Der Shieh, Yin-Tsung Hwang, Hanho Lee, Chirn Chye Boon, and Zhiyuan Yan
Editorial Board

The editorial board of the journal is organized into sections that correspond to the subject areas covered by the journal.

Circuits and Systems

M. T. Abuelma’atti, Saudi Arabia
Ishfaq Ahmad, USA
Dhamin Al-Khalili, Canada
Wael M. Badawy, Canada
Ivo Barbi, Brazil
Martin A. Brooke, USA
Chip Hong Chang, Singapore
Y. W. Chang, Taiwan
Tian-Sheuan Chang, Taiwan
Tzi-Dar Chiueh, Taiwan
Henry S. H. Chung, Hong Kong
M. Jamal Deen, Canada
Ahmed El Wakil, UAE
Denis Flandre, Belgium
P. Franzon, USA
Andre Ivanov, Canada
Ebroul Izquierdo, UK
Wen-Ben Jone, USA
Yong-Bin Kim, USA
H. Kuntman, Turkey
Parag K. Lala, USA
Shen-Iuan Liu, Taiwan
Bin-Da Liu, Taiwan
João Antonio Martino, Brazil
Panki Mazumder, USA
Michel Nakha, Canada
Sing Kiong Nguang, New Zealand
Shun-ichiro Ohmi, Japan
Mohamed A. Osman, USA
Ping Feng Pai, Taiwan
Marcelo Antonio Pavanello, Brazil
Marco Platzner, Germany
Massimo Poncino, Italy
Dhiraj K. Pradhan, UK
F. Ren, USA
Gabriel Robins, USA
Mohamad Sawan, Canada
Raj Senani, India
Gianluca Setti, Italy
Jose Silva-Martinez, USA
Ahmed M. Soliman, Egypt
Dimitrios Soudris, Greece
Charles E. Stroud, USA
Ephram Suhir, USA
Hannu Tenhunen, Sweden
George S. Tombras, Greece
Spyros Tragoudas, USA
Chi Kong Tse, Hong Kong
Chi-Ying Tsui, Hong Kong
Jan Van der Spiegel, USA
Chin-Long Wey, USA

Communications

Sofiène Affes, Canada
Dharma Agrawal, USA
H. Arslan, USA
Edward Au, China
Enzo Baccarelli, Italy
Stefano Basagni, USA
Jun Bi, China
Z. Chen, Singapore
René Cumplido, Mexico
Luca De Nardis, Italy
M.-G. Di Benedetto, Italy
J. Fiorina, France
Lijia Ge, China
Zabih F. Ghassemlooy, UK
K. Giridhar, India
Amoakoh Gyasi-Agyei, Ghana
Yaohui Jin, China
Mandeep Jit Singh, Malaysia
Peter Jung, Germany
Adnan Kavak, Turkey
Rajesh Khanna, India
Kiseon Kim, Republic of Korea
D. I. Laurenson, UK
Tho Le-Ngoc, Canada
C. Leung, Canada
Petri Mähönen, Germany
M. Abdul Matin, Bangladesh
M. Nájar, Spain
Mohammad S. Obaidat, USA
Adam Panagos, USA
Samuel Pierre, Canada
John N. Sahalos, Greece
Christian Schlegel, Canada
Vinod Sharma, India
Iickho Song, Republic of Korea
Ioannis Tomkos, Greece
Chien Cheng Tseng, Taiwan
George Tsoulos, Greece
Laura Vanzago, Italy
Roberto Verdone, Italy
Guosen Yue, USA
Jian-Kang Zhang, Canada

Signal Processing

S. S. Agaian, USA
P. Agathoklis, Canada
Jaakko Astola, Finland
Tamal Bose, USA
A. G. Constantinides, UK
Paul Dan Cristea, Romania
Petar M. Djuric, USA
Igor Djurović, Montenegro
Karen Egiazarian, Finland
W. S. Gan, Singapore
Zabih F. Ghassemlooy, UK
Ling Guan, Canada
Martin Haardt, Germany
Peter Handel, Sweden
Andreas Jakobsson, Sweden
Jiri Jan, Czech Republic
S. Jensen, Denmark
Chi Chung Ko, Singapore
M. A. Lagunas, Spain
J. Lam, Hong Kong
D. I. Laurenson, UK
Riccardo Leonardi, Italy
Mark Liao, Taiwan
Stephen Marshall, UK
Antonio Napolitano, Italy
Sven Nordholm, Australia
S. Panchanathan, USA
Periasamy K. Rajan, USA
Cdrich Richard, France
William Sandham, UK
Ravi Sankar, USA
Dan Schonfeld, USA
Ling Shao, UK
John J. Shynk, USA
Andreas Spanias, USA
Srdjan Stankovic, Montenegro
Yannis Stylianou, Greece
Ioan Tabus, Finland
Jarmo Henrik Takala, Finland
A. H. Tewfik, USA
Jitendra Kumar Tugnait, USA
Vesa Valimaki, Finland
Luc Vandendorpe, Belgium
Ari J. Visa, Finland
Jar Ferr Yang, Taiwan
Contents

Implementations of Signal-Processing Algorithms for OFDM Systems, Ming-Der Shieh, Yin-Tsung Hwang, Hanho Lee, Chirn Chye Boon, and Zhiyuan Yan
Volume 2012, Article ID 687172, 2 pages

VLSI Architectures for Sliding-Window-Based Space-Time Turbo Trellis Code Decoders, Georgios Passas and Steven Freear
Volume 2012, Article ID 614259, 14 pages

Interference Cancellation Using Replica Signal for HTRCI-MIMO/OFDM in Time-Variant Large Delay Spread Longer Than Guard Interval, Yuta Ida, Chang-Jun Ahn, Takeshi Kamio, Hisato Fujisaka, and Kazuhisa Haeiwa
Volume 2012, Article ID 571985, 10 pages

A Power-Efficient Soft-Output Detector for Spatial-Multiplexing MIMO Communications, Hsiao-Chi Wang, Tung-Lin Liu, Yuan-Wei Wu, and Hsi-Pin Ma
Volume 2012, Article ID 938490, 9 pages

CP-Based SBHT-RLS Algorithms for Tracking Channel Estimates in Multicarrier Modulation Systems, H. Ali
Volume 2012, Article ID 452806, 12 pages

Two-Dimensional Iterative Processing for DAB Receivers Based on Trellis-Decomposition, Wim J. van Houtum and Frans M. J. Willems
Volume 2012, Article ID 394809, 15 pages

Advanced Receiver Design for Mitigating Multiple RF Impairments in OFDM Systems: Algorithms and RF Measurements, Adnan Kiayani, Lauri Anttila, Yaning Zou, and Mikko Valkama
Volume 2012, Article ID 730537, 16 pages

Loop-Reduction LLL Algorithm and Architecture for Lattice-Reduction-Aided MIMO Detection, Chun-Fu Liao, Li-Wei Chai, and Yuan-Hao Huang
Volume 2012, Article ID 876380, 7 pages

Probabilistic Behavior Analysis of MIMO Fading Channels under Geometric Mean Decomposition, Ping-Heng Kuo and Pang-An Ting
Volume 2012, Article ID 340809, 8 pages
Orthogonal frequency-division Multiplexing (OFDM) techniques, coupled with multiple-input multiple-output (MIMO) processing, have been widely deployed in various wireless communication systems such as 802.11n, WiMAX, and 3GPP LTE. To enhance the system performance, novel signal processing techniques are developed, which often require massive and complicated computations and pose major challenges in implementations. With the advances of VLSI technology, the employment of these sophisticated signal processing algorithms now becomes feasible via dedicated mapping onto powerful digital signal processors or dedicated hardware modules. Joint consideration of algorithm development and implementation issues is crucial to achieve optimized solutions. This special issue is created to bring together the state-of-the-art research contributions to the digital signal processing techniques tailored for OFDM systems. It has indeed successfully attracted the submissions of many high-quality papers. After going through a conscientious peer-review process, eight papers are selected. These papers make an inspiring ensemble with a topic spectrum ranging from novel algorithm development, theoretical system characterization, to efficient hard-wired implementations in FPGA/ASIC. Although research topics on signal processing for MIMO-OFDM systems are too broad to be covered in one special issue, we believe that the collected papers dutifully serve as a good starting point. Finally, we would like to thank the authors for their excellent contributions and the cooperation during the editorial process. The competent and timely work by all the reviewers of these papers is also highly appreciated.

For the eight papers included in this special issue, two of them aim at channel estimation and tracking. Another two papers work on the MIMO signal detection schemes. Still another two papers are related to Trellis decoder designs. We also have one paper focusing on the mitigation of RF impairments and another one provides probabilistic behavior analysis of MIMO channels.

Specifically, in the paper entitled “Probabilistic behavior analysis of MIMO fading channels under geometric mean decomposition,” P. H. Kuo and P. A. Ting presents statistical characterization of link gains and channel capacity that can be achieved by using a GMD precoding scheme. Problem formulation, PDF and channel capacity derivation, finite-state Markov chains constructions, and simulation results are provided with satisfactory performance.

In the paper entitled “CP-based SBHT-RLS algorithms for tracking channel estimates in multicarrier modulation systems” by H. Ali, two new algorithms that exploit cyclic prefix for data detection and employ systolic block Householder transformation recursive least squares (SBHT-RLS) algorithms for channel tracking are presented. The new methods outperform existing RLS-based channel tracking schemes. Computational complexity and parallel implementation issues are also addressed. Complete algorithm performance evaluations subject to varying forgetting factor parameter values, constellation size, and word lengths are provided to illustrate the advantages of the proposed work.

The paper is entitled “Advanced receiver design for mitigating multiple RF impairments in OFDM systems algorithms and RF measurements.” Kiayani et al. investigate the joint
impact of frequency-selective I/Q imbalance at both trans-
mitter and receiver together with channel distortions and
CFO errors. Two estimation and compensation structures
based on different pilot patterns are proposed to separate
the individual impairments. Low-complexity time domain
estimation and compensation algorithms are next developed
to mitigate the problems. Both simulation and measurement
results support their performance claims.

There are two papers both working on the topic of
MIMO-OFDM signal detection. In the paper entitled “Loop-
reduction LLL algorithm and architecture for lattice-reduction-
aided MIMO detection,” Liao et al. propose a loop reduction
LLL algorithm and its architecture for lattice-reduction-
aided MIMO detection. A look-ahead check technique is
applied to reduce the algorithm complexity while observing
the original LLL criterion. Simulation results show that not
only the computation complexity but also the computing
latency are effectively reduced.

In the paper “Two-dimensional iterative processing for
DAB receivers based on trellis-decomposition,” W. J. Houtum
and F. M. J. Willems investigate trellis decoding and iterative
techniques for digital audio broadcasting (DAB). Trellis-
decomposition methods allow us to estimate the unknown
channel phase, since this phase relates to the subtrellises.
Simulations of this iterative approach applied to the TU-
6 (COST207) channel show an improvement of 2.4 dB at a
Doppler frequency of 10 Hz.

In the paper entitled “A power-efficient soft-output
detector for spatial-multiplexing MIMO communications,”
Wang et al. propose VLSI implementation of a configurable
power-efficient MIMO detector that supports 4 × 4 spatial
multiplexing and modulation from QPSK to 64-QAM. A
novel tree search algorithm is also proposed to enable the
detector to provide soft outputs and to be implemented in
parallel and pipelined hardware architecture.

The paper entitled “Interference cancellation using replica
signal for HTRCI-MIMO/OFDM in time-variant large delay
spread longer than guard interval” focuses on the time domain
ISI compensation through the replica signal insertion with
the shifted channel impulse response. System models and
complexity analysis are also provided. Exhaustive simula-
tions are conducted to show their performance edge.

In the paper entitled “VLSI architectures for sliding-
window-based space-time turbo trellis code decoders,” G.
Passas and S. Freear present VLSI architectures of space-time
turbo trellis coding decoders as well as of a set of SISO-MAP
component channel decoders used in turbo coding based
on sliding window. They consider space-time turbo-code
receivers based on nonbinary trellises. Finally, measurements
of complexity and decoding delay of several sliding window
technique-based MAP decoder architectures are developed
and compared.

Ming-Der Shieh
Yin-Tsung Hwang
Hantho Lee
Chirn Chye Boon
Zhiyuan Yan
The VLSI implementation of SISO-MAP decoders used for traditional iterative turbo coding has been investigated in the literature. In this paper, a complete architectural model of a space-time turbo code receiver that includes elementary decoders is presented. These architectures are based on newly proposed building blocks such as a recursive add-compare-select-offset (ACSO) unit, A-, B-, Γ-, and LLR output calculation modules. Measurements of complexity and decoding delay of several sliding-window-technique-based MAP decoder architectures and a proposed parameter set lead to defining equations and comparison between those architectures.

1. Introduction

The ability of turbo codes (TCs) to achieve very low BER that approaches the Shannon limit is very attractive. These channel capacity approaching codes have been proposed by Berrou et al. [1, 2]. Iterative turbo decoding involves the parallel concatenation of two or more component codes, which operate on the data and exchange information in order to progressively reduce the error rate. The exchanged information is in the form of log-likelihood ratio (LLR) soft decisions and can be exchanged between elementary decoders, which apply either the soft output Viterbi algorithm (SOVA) [3, 4] or the maximum a posteriori (MAP) probability algorithm [5].

The principles of iterative turbo decoding can be combined with those of space-time coding resulting in a bandwidth efficient low-error-rate channel coding scheme named space-time turbo trellis coding (STTuTC) [6–10]. The latter scheme benefits from the impressive coding gain of turbo codes and the diversity gain of space-time codes, to obtain a very power-efficient system. These power gains are very important for high-performance communication systems, particularly in scenarios where low signal-to-noise ratio is in demand.

Despite the complexity reduction, iterative turbo codes still have prohibitively high implementation complexity and suffer from large decoding delay, motivating researchers to seek efficient implementations [11–19]. The sliding window technique (SWT) presented in [17–21] enables the early acquisition of the state metric values without having to scan the whole trellis frame in the one direction before scanning the other, and this results in reduced elementary decoder latency as well as smaller state metric memory requirements.

2. Architectural Overview

2.1. Turbo Transmitter. A space-time turbo trellis code is a parallel concatenation scheme using two identical elementary encoders to operate on an information message $C = [C_0, C_1, \ldots, C_{N-1}, C_P]$, where each symbol consists of $b$ bits [8, 9]. The same message is sent to both component encoders, but each one receives this information in a different order through an interleaver.

Elementary encoders using recursive space time trellis coding (STTC) benefits from interleaver gain and iterative decoding [6–8]. These encoders add redundancy to the encoded messages, and the second sequence is deinterleaved back to its original order. The two sequences of codewords
are multiplexed and truncated to preserve bandwidth efficiency and form a single $P$ length sequence denoted by $X = [X_0, X_1, \ldots, X_t, \ldots, X_{p-1}]$, where each space-time vector $X_t = [x_{1,t}, x_{2,t}, \ldots, x_{1,p}, x_{2,p}, \ldots, x_{K,t}]^T$ consists of $K$ $b$-bit modulated symbols. In vector $X$, half of the symbols (odd locations) come from component encoder 1 ($p = 1$) and the other half (even locations) come from component encoder 2 ($p = 2$) [6]. The sequence is forwarded for carrier modulation and transmission since baseband modulation occurs within the encoders. The structure results in a long block code from small memory convolutional codes, a fact that enables the turbo code to approach the Shannon limit.

Note that the recursive STTCs are trellis based, meaning that during the encoding process, in each elementary encoder a trellis path is created [7]. Let $S = \{s_0, s_1, \ldots, s_t, \ldots, s_{p-1}, s_p\}$ be a state row vector containing all the states in a trellis path chosen by the information message. The combined trellis paths will be decomposed and exploited at the receiver side to decode the message.

2.2. Proposed Turbo Receiver. Consider a baseband MIMO receiver, shown in Figure 1, the received frame $R = [R_0, R_1, \ldots, R_t, \ldots, R_{p-1}]$, where $R_t = [r_{1,t}, r_{2,t}, \ldots, r_{1,t}, r_{2,t}, \ldots, r_{K,t}]^T$, can be used to estimate the original sequence $C$. Using the baseband signal representation, the received noisy signal from antenna $j$ at time $t$ over a MIMO (possibly fading, when $h_{j,i} \neq 1$) Gaussian channel is given by the equation

$$r_{j,t} = V_{j,t} + n_{j,t} = \sum_{i=1}^{K} h_{j,i} x_{i,t} + n_{j,t}, \quad (1)$$

where $V_{j,t}$ represents the noiseless version of the $r_{j,t}$ signal [7–9]. The receiver, after estimating the channel fading coefficients, separates the signals using maximal ratio receiver combining (MRRC), which also equalises the signals by removing the effect of the channel. With MRRC, costly joint detection of space-time symbols is avoided.

The equalised frame symbols are delivered to a soft-demapper to calculate the soft channel output given in

$$H_j(s', s) = \ln(\eta_j(s', s)) = \ln\left(\frac{e^{-E_r||R_j - V_j||^2/2\sigma^2}}{2\sigma^2}\right) \quad (2)$$

The received symbols have already $-E_r/2\sigma^2$ prescaled by a SNR scaler in the analogue domain [12]. A frame formation block (FFB) separates the even and odd symbols and inserts zeros, so it is assured that each log-MAP decoder core receives information only from the corresponding component encoder.

The STTuTC decoder at the algorithmic level consists of two soft-in soft-out (SISO) component decoders, which exchange soft information and progressively through many iterations result in a better estimate of the values of the information symbols [11, 12]. Since the two component decoders do not operate on data at the same time, only one silicon IP core can be used to implement both. As illustrated in Figure 1, the decoder contains a symbol-based SISO component core that applies the log-MAP algorithm on a nonbinary trellis. It also uses two memories, one for storing the exchanged soft information and another to perform symbol interleaving/deinterleaving.

2.3. MAP and Log-MAP Algorithms. Maximum likelihood decoding using Viterbi algorithm determines the most likely path through the trellis and from that determines the

Figure 1: VLSI architectural model of STTuTC receiver.
information symbol sequence, whereas MAP decoding determines the latter sequence by considering each information symbol $c_t$ independently using the channel observations $R$. The MAP algorithm is used to compute the a posteriori LLR soft decisions (3), about the value of the symbols in the information message [6, 8]:

$$\lambda^f(u_t||R) = \ln \left( \frac{\Pr(u_t = f||R)}{\Pr(u_t = 0||R)} \right) = \ln \left( \frac{\sum_{s' \in A}(b_t(s))y^f_t(s', s)a_{t-1}(s')}{\sum_{s' \in A}(b_t(s))y^f_t(s', s)a_{t-1}(s')} \right),$$

(3)

where $f = 0, 1, 2, \ldots, 2^b - 1$. This equation expresses the excess probability of a symbol $u_t$ being $f$ in the logarithmic domain over the probability of being equal to a reference symbol $0$ [11, 13]. The log-MAP decoder calculates the LLRs of all the possible modulation symbols in the constellation diagram using the forward $A_t(s) = \ln(a_t(s))$ and backward $B_t(s) = \ln(b_t(s))$ state metric probabilities as well as the branch metric probabilities $\Gamma_t(s', s) = \ln(\gamma^f_t(s', s))$ in the logarithmic domain:

$$A_t(s) = \sum_{s' = 0}^{2^t - 1} \sum_{f = 0}^{2^t - 1} \left( \Gamma^f_t(s', s) + A_{t-1}(s') \right),$$

$$B_{t-1}(s') = \sum_{s = 0}^{2^t - 1} \sum_{f = 0}^{2^t - 1} \left( \Gamma^f_t(s', s) + B_t(s) \right).$$

(4)

The $A_t(s)$ metric represents the logarithmic probability that the state $s$ at stage $t$ has been reached from the beginning of the trellis. Similarly, the $B_{t-1}(s')$ metric represents the logarithmic probability that the state $s'$ at stage $t - 1$ has been reached from the end of the trellis [13, 15]. The branch metric probabilities can be computed using

$$y^f_t(s', s) = \Pr(R_t \cap s\mid s') = \Pr(R_t || s \cap s') \Pr(s||s') \Rightarrow$$

$$y^f_t(s', s) = \Pr(R_t || V_t) \cdot \frac{\Pr(f)}{\Pr(0)} = \frac{1}{2\pi\sigma^2} e^{-|R_t - V_t|^2/2\sigma^2} \cdot \frac{\Pr(f)}{\Pr(0)} \Rightarrow$$

$$y^f_t(s', s) = C_t e^{-|R_t - V_t|^2/2\sigma^2} \cdot \frac{\Pr(f)}{\Pr(0)} = C_t \eta_t(s', s) \cdot \frac{\Pr(f)}{\Pr(0)},$$

(5)

where $C_t = 1/2\pi\sigma^2$. The above equation gives the probability of a transition at time $t$ from state $s'$ to state $s$ in the trellis diagram [13]. Translating (5) in the logarithmic domain [15],

$$\Gamma^f_t(s', s) = \ln(y^f_t(s', s)) = \tilde{C}_t + H_t(s', s) + \ln \left( \frac{\Pr(f)}{\Pr(0)} \right),$$

(6)

where $\tilde{C}_t = \ln(C_t)$ is a constant, the $H_t(s', s)$ term is the soft channel output, and the $\ln(\Pr(f)/\Pr(0))$ term is the a priori information coming from the other component decoder:

$$\lambda^f(u_t||R) = \ln \left( \frac{\sum_{s'}(s' \in A)(b_t(s))y^f_t(s', s)a_{t-1}(s')} {\sum_{s'}(s' \in A)(b_t(s))y^f_t(s', s)a_{t-1}(s')} \right) \Rightarrow$$

$$\lambda^f(u_t||R) = \ln \left( \sum_{s'}(s' \in A)(b_t(s))y^f_t(s', s)a_{t-1}(s') + A_{t-1}(s') \right)$$

$$- \ln \left( \sum_{s'}(s' \in A)(b_t(s))y^f_t(s', s)a_{t-1}(s') \right) \Rightarrow$$

$$\lambda^f(u_t||R) = \max_{s'}(B_t(s) + \Gamma^f_t(s', s) + A_{t-1}(s')).$$

(7)

Finally, (7) gives the elementary decoder output, which is a posteriori probability LLR in terms of logarithmic domain state and branch metrics [13].

3. Soft-In Soft-Out-MAP Component Decoder

3.1. Decoder Module Architecture. The proposed elementary decoder core is illustrated in Figure 2. The main modules within this decoder are the branch metric calculation module (BM-CM), the A- and B-state metric calculation modules (A-SM-CM and B-SM-CM, resp.), the LLR calculation module (LLR-CM), and a large A-metric RAM to temporarily store metric values.

The BM-CM calculates all possible normalised branch metric values (see Section 3.1) and store them in a register file. The latter consists of $M = 2^b$ registers to store the $M$ possible branch metric values. These metrics are given to A-SM-CM, B-SM-CM, and LLR-CM. The A-metric CM scans the entire trellis diagram in the forward direction, starting from stage zero and ending at stage $P$, calculating all A-state metrics and storing them in the A-metric RAM.

After this first trellis pass the B-metric CM and the LLR-CM simultaneously compute the B-state metrics and the LLR outputs, respectively. The B-SM-CM scans the entire trellis in the backward direction, from stage $P$ to 0. At this time, the B-metrics do not need to be stored in a memory, since every time a B-metric is calculated the corresponding A-metric is extracted from the A-metric RAM and the LLR calculation is computed.

Therefore, only the A-state metrics are stored in a RAM memory, the size of which must be $(P + 1) \times 2^G \times w_t$, for continuous mode decoders. In a decoder operating in the terminated mode, where the trellis path begins and ends at state $s'(0)$, that is, $s_0 = s(0)$ and $s_P = s(0)$, the memory requirement is slightly reduced to $((P - 3) \times 2^G + 2 + 2^b + 1) \times w_t$, whereas in truncated operating mode (i.e., $s_0 = s(0)$ must hold for every frame, but there is no termination condition), the storage requirement is $((P - 1) \times 2^G + 2^b + 1) \times w_t$. Where $w_t$ is the word length of the data format.
used to represent the internal data of the MAP decoder. Table 1 shows the complexity and decoding delay of a log-MAP decoder operating in different modes. The metrics are calculated using data flow units (DFUs) that operate in decoding steps. As can be seen the log-MAP decoder results in very high memory requirements and long elementary decoder latencies.

3.2. Branch Metric Calculation Module. The branch metric calculation module computes the normalised branch metrics $\Gamma^f_t(s', s)$. The branch metric appears on both the numerator and denominator of the a posteriori LLR formula, thus any constants will cancel out and play no role in the calculation:

$$\Gamma^f_t(s', s) = H_t(s', s) + \ln \left( \frac{P_{t}(f)}{P_{t}(0)} \right), \quad (8)$$

The module receives as input the soft output of the channel $H_t(s', s)$ and the a priori information from the other component decoder and computes all possible values of branch metrics as shown in Figure 3.

3.3. State Metric Calculation Module. In the literature, the proposed architectures that recursively calculate the state metrics are based in the so-called ACSO (add-compare-select-offset) processing unit equivalent to add-max*() presented in [18]. It is responsible for the computation of a state metric based on the previous stage metrics and the branch metrics arriving at the state in question.

Several different, but similar architectures have been proposed. The diagram in Figure 4 suggests a new type of the ACSO unit, which instead of employing more multiplexers to cope with the more than two input state metrics in nonbinary trellises, it recursively accepts the input state and branch metrics and calculates the output state metric in a number of cycles. A look-up table can be used to store precalculated correction factors and its quantization is discussed in [18, 22].

This module requires $M$ clock cycles to calculate the output state metric. In the first clock cycle, the feedback register is reset, and the $A_{t-1}(s'_t)$ and $\Gamma^f_{t-1}(s'_t, s)$ are applied at the inputs. Also the SW1 switch is connected to the zero
The result is that $a_1 = A_{t-1}(s'_1) + \Gamma_0^t(s'_1, s)$ is stored in the feedback register. In the remaining cycles, the LUT output is connected through SW1 to the output adder. In the second cycle, the $a_2 = A_{t-1}(s'_2) + \Gamma_1^t(s'_2, s)$ is applied at the input; therefore the $\max^*(a_1, a_2)$ is stored in the feedback register. Similarly, the $\max^*(a_3, \max^*(a_1, a_2))$ is stored at the third cycle, whereas the output state metric equal to $\max^*(a_4, \max^*(a_3, \max^*(a_1, a_2)))$ is ready at the output port during the forth clock cycle. A-metric calculation is illustrated, but the architecture operates equally good for B-metric calculation in a backward recursion. This unit will be referred to as recACSO unit.

The forward/backward recursion computations may lead to an over flow. If a state metric value constantly grow the finite word length $w_L$ will not be sufficient to hold this value and an over flow will occur. Since the $\max^*(\cdot)$ operation is linear and shift invariant and a global shift of state metric values would not change the LLR output value, it is the difference between the state metrics and not their absolute values that is important. Rescaling of the state metrics can be readily performed to avoid over flow. The rescaling technique is the same as that used in other SISO-log MAP algorithms [11, 18, 23].

The precision of the state metrics determines the word length $w_L$. The precision depends on the dynamic range of the state metrics.

Thus, $2^G$ recACSO units can be employed to calculate all the states of trellis stage in one step, as depicted in Figure 5 for a $2^G$ state trellis. This unit receives $2^G$ possible branch metrics and the previous state metrics and computes all the state metrics of the current stage. Configuring the module this way fastens computation since it avoids reading the previous
stage state metrics from the RAM memory. At the input of the module two selectors are used to drive the recACSO units with the appropriate inputs.

3.4. LLR Output Calculation Module. The LLR-CM depicted in Figure 6 is responsible for computing the output reliability estimates $\lambda^f(u_t \mid R)$ given the state and branch metric calculations according to

$$\lambda^f(u_t \mid R) = \max_{(s', s)} \left( B_t(s) + \Gamma^f_t(s', s) + A_{t-1}(s') \right)$$

$$- \max_{(s', s)} \left( B_t(s) + \Gamma^0_t(s', s) + A_{t-1}(s') \right).$$

(9)
A careful observation of (9) reveals a close relation with the recACSO unit calculation. The LLR computation has two terms that can be computed separately and then subtracted. If $A_{t-1}(s')$ and $B_t(s)$ are added in advance, then the recACSO unit can then be used to calculate $\text{max}^*(f)$, where $f = 0, 1, \ldots, 2^b - 1$. Because the $\text{max}^*(f)$ when $f = 0$ has to be subtracted from all other LLRs, an inverter is included to negate this term.

As already mentioned, $\lambda_0(u_t \mid R)$ need not be calculated since zero is the reference symbol and therefore its LLR is zero. The remaining $2^b - 2$ log-likelihood ratios are computed by the proposed architecture and outputted out of the symbol-based log-MAP component decoder.

4. SWT-Based Architectures for the SISO-MAP Decoder

4.1. The Concept. SISO-MAP decoders suffer from very long latency and high memory requirements. A different organisation (scheduling) of the computation and exploitation of the convergence property [24–26] of the decoder lead to reduction in the latency and the amount of state metrics that need to be stored. For this reason, the sliding window technique (SWT) has been proposed in the literature [18, 19].

SWTs are divided in single-flow structure (SFS), double-flow structure (DFS), and pointer-based (PNT) techniques. In these techniques the trellis or transmitted frame is divided in windows of size $L$, where $L$ is the convergence length [25]. Care must be taken to ensure that the whole frame can be exactly divided into an integer number of windows. MAP decoder convergence property allows recursive calculation of a state metric to result in a good approximation of the actual metric value after $L$ recursion steps, even if the initial data is a dummy value [8, 9].

Three important metrics for comparison of MAP decoders can be considered: (i) decoding delay (latency), (ii) memory requirements (or storage lifetime), and (iii) computational complexity (number of recursion modules). Several different schedulings of the MAP algorithm computation reveal trade-offs between these quantities. Since the window size is $L$ recursion steps, a DFU proposed in Section 3 can be employed to calculate all the state metric vectors in

---

**Figure 8:** State metric calculation modules for single-flow structures.

**Figure 9:** DFG for double-flow structures (DFS, $\Pi = L, \pi = 1/2$, shift = $L/4$).
a window in $L$ steps. A DFU can perform forward, backward, or dummy metric calculations.

Defining the following parameter set will assist in the analysis of SWT-based MAP decoders (technique, $\Pi$, $\pi$, shift, $\varepsilon$). The “technique” element is the type of computation organisation with possible values being SFS, DFS, SFS/PNT, and DFS/PNT.

$\Pi$ is the relative position between (A- and B-) forward and backward coupled metric recursive calculations and therefore determines when the B-metric calculation begins relative to A-metric calculation. It is a continuous value variable, but only three values are a meaningful choice. $\Pi = 0$ means that the backward calculation starts after the forward ends, $\Pi = 2L$ means that the forward calculation starts after the backward ends, and $\Pi = L$ means that the two calculations take place simultaneously and switch role in the middle of the convergence block calculation.

$\pi$ is the ratio of the valid over the invalid (dummy) metric recursive calculation. To compute the backward metrics before the end of the whole frame a dummy metric calculation is required to estimate an intermediate value using the convergence property. This means that for every A- and B-metric pair one dummy module is required, but that can vary depending on the technique, $\pi$ is the ratio of invalid metrics over valid A- and B-metrics.

The computation can take place in two directions (two flows) simultaneously each calculating half of the frame. The “Shift” is the time shift in decoding steps in between the two flows. In the pointer technique pointers are used to reduce memory requirements, $\varepsilon$ is the number of pointers, which when exists indicates the pointer technique PNT.

This set of SWT-based architectures has been fully investigated and some results are given in Table 2. These indicate the hardware requirements and decoding delay of some common implementation structures.
4.2. Single-Flow Structure SISO-MAP Decoders. Figures 7 and 8 illustrate, as an example, the data flow graph (DFG) of a single flow and the architecture of SM-CM of \((SFS, \Pi = 0, \pi = 1)\) and the architecture of \((SFS, \Pi = 0, \pi = 1/2)\) single-flow structures. The flow diagram explains the computation.

In the DFG the horizontal axis represents time quantized in terms of symbol periods, whereas the vertical axis represents the symbol number within the frame. In this specific diagram a frame of \(P = 4L\) symbols decoded by a trellis of \(P + 1 = 4L + 1\) stages is presented, where \(L\) is the convergence length of the decoder.

Three data flow units can be seen in the diagram. The dashed arrow represents \(L\) recursions of a forward DFU calculating A-metric, which are stored as calculated. The continuous arrow represents \(L\) recursion of a backward DFU calculating B-metrics. This calculation is done simultaneously with the LLR soft output calculation. The dotted arrow represents \(L\) recursions of a dummy metric DFU.

In the diagrams to follow the dotted arrow always represents dummy metric recursive use of DFUs, whereas for forward and backward metric DFUs the arrow is continuous or dashed when the metric is not stored or stored in the memory, respectively. The dummy metric DFU results in a valid metric after the \(L\) backward recursions.

Storage is also represented by the shaded rectangular areas. Taking into account one of these areas (they are all the same), the projection of this rectangular area on the vertical line gives the amount of state metric vectors to be stored \([18, 21]\), whereas the projection on the horizontal line gives the time required to store the vectors.

Decoding delay (or latency) is the horizontal distance between the acquisition curve (always \(y_1 = t\)) and the decoding curve in DFG. Since the decoding curve in this structure is at \(y_2 = t - 4L\), the decoding delay is \(y_1 - y_2 = t - (t - 4L) = 4L\) symbol periods. The symbol period is denoted by \(t_s\).

DFGs and tile graphs model the same thing, they model the resource-time scheduling of the recursions of the algorithm. A diagram is a DFG when viewed as a concrete graph or tile graph when viewed as tile repetition \([17]\). On the right of Figure 7 a tile, which consist of 3 DFUs, is illustrated. This tile is repeated as many times as required to form a complete DFG.

Let us concentrate more on the scheduling of the operations described by the DFG in Figure 7. During the symbol periods \(2L\) to \(3L - 1\), the dummy metric DFU (dotted arrow 1) starting from a zero vector assigned to state metric vector \(B_L\) computes invalid metrics from \(B_{2L-1}\) down to \(B_{L}\). This calculation results in a valid state metric vector \(B_L\), because convergence is reached. All other metrics calculated during these \(L\) recursions are invalid. At the same time the forward DFU (dashed arrow 2) calculates and stores a sequence of \(L\) A-metrics from \(A_0\) to \(A_{L-1}\).

In the next \(L\) symbol periods from \(3L\) to \(4L - 1\) the backward DFU (continuous arrow 3) comes into play to
calculate $B_{L-1}$ to $B_0$. During the backward calculation the A-state metrics from $A_{L-1}$ to $A_0$ are extracted from the memory and together with the B-state metrics are used to calculate the first $L$ soft outputs.

Thus, a set of 3 recursion units (called a “tile” in a tile diagram as indicated in DFG) results in the calculation of $L$ A- and B-metrics required to produce $L$ soft output LLR values. Note that the LLR values are calculated in a reverse manner. At symbol periods $4L$ to $5L-1$, the order of soft output values is reversed [18]. In a turbo coding scheme, the interleaver reordering can be exploited to perform this reversing. If this last LLR reversing step is done by the MAP decoder, then the latency of this core is $4L$ symbol periods. If the reversing is done by the interleaver, the decoding delay is $3L$ symbol periods.

The above process (tile) is repeated until the end of the frame. The set of tiles in the tile graph (or DFG) represents a single flow passing through the data of a frame (or trellis). Hence, the proposed structure is designated as a single-flow structure (SFS). Structures where the above process occurs in both directions are called double-flow structures (DFSs) and will be discussed later in this paper.

In single-flow structures the benefit of structures with $\Pi = 0$ compared to $\Pi = L$ is the use of a single RAM memory.

**Figure 13:** VLSI Architecture for SWT-based MAP decoder using the double-flow structures and the pointer technique.
as opposed to two RAM memories of half size each, required in the latter structures. Although the memory requirements as well as other requirements are the same, less complex datapath circuits are needed in the (SFS, \( \pi = 0 \)) case.

Also the (SFS, \( \pi = L \)) structure experiences smaller decoding delay than the (SFS, \( \pi = 2L \)) case. LLR reversing is performed in the interleaver rather than in the MAP decoder. Thus, for SFS, \( \pi = 0 \) is clearly the best choice. For (SFS, \( \pi = 2L \)) decoders,

\[
\text{storage req. [bits]} = \pi L^2 w_L,
\]

\[
\text{number of DFUs} = \begin{cases} 
3, & \pi \geq 1, \\
2 + \frac{1}{\pi}, & \pi \leq 1,
\end{cases}
\]

\[
\text{decoding delay} = \begin{cases} 
3\pi L & \text{or } 2\pi L, \\
(2 + 2\pi)L & \text{or } (2 + \pi)L, \\
\end{cases}
\]

For \( \pi < 1 \), the storage requirements and decoding delay are decreasing as \( \pi \) decreases, but the number of required DFUs is increasing. The decoding delay is \((2 + 2\pi)L\) when MAP decoder reverses the LLR outputs and \((2 + \pi)L\) when the interleaver reverses the LLR outputs.

### 4.3 Double-Flow Structure SISO-MAP Decoders

The DFG and the state metric calculation modules for a (DFS, \( \pi = L \)) structure are shown in Figures 9 and 10, respectively. The double-flow structures use all of the concepts of single flows, in the sense that they also have parameters \( \pi \) with the interesting values being \( \pi = 0 \), \( L \) and \( 2L \) and \( \pi = 1/2, 1 \) and 2. Some DFSs have the benefit of operating with twice as much speed, without doubling the hardware requirements.

For example, when \( \pi = 1 \) the RAM memories used by the data flow units to store the state metrics A and B of the SISO-MAP algorithm operate in the write mode for half of the time. The remaining half of the time they operate in read mode. This read period can be exploited by a second flow which runs over half of the frame data in the opposite direction. This is illustrated in Figure 9 for \( \pi = 1 \), which shows a structure that decodes a frame of \( \pi = 8L \) symbols in half time.

### Table 2: Complexity and decoding delay of SWT-based log-MAP decoders.

| Parameter set | Mem. req. [bits] | DFUs | Decoding delay          |
|---------------|------------------|------|-------------------------|
| (SFS, \( \pi = 0 \)) | \( L^2 w_L \) | 3    | 4L/3L                   |
| (SFS, \( \pi = L \)) | \( L^2 w_L \) | 3    | 4L/3L                   |
| (SFS, \( \pi = 2L \)) | \( L^2 w_L \) | 3    | 4L/4L                   |
| (SFS, \( \pi = 0, \pi = 1/2 \)) | \( (L/2)^2 w_L \) | 4    | 3L/2.5L                 |
| (SFS, \( \pi = 0, \pi = 1/3 \)) | \( (L/3)^2 w_L \) | 5    | 2.66L/2.33L             |
| (SFS, \( \pi = 0, \pi = 1/4 \)) | \( (L/4)^2 w_L \) | 6    | 2.5L/2.25L              |
| (SFS, \( \pi = 0, \pi = 2 \)) | \( 2L^2 w_L \) | 3    | 6L/4L                   |
| (SFS, \( \pi = 0, \pi = 3 \)) | \( 3L^2 w_L \) | 3    | 9L/6L                   |
| (SFS, \( \pi = 0, \pi = 4 \)) | \( 4L^2 w_L \) | 3    | 12L/8L                  |
| (DFS, \( \pi = 0, \pi = 1, \text{shift}=0 \)) | \( 2L^2 w_L \) | 6    | 4L/3L                   |
| (DFS, \( \pi = L, \pi = 1 \text{ shift}=L/2 \)) | \( L^2 w_L \) | 6    | 4.5L/3.5L               |
| (DFS, \( \pi = L, \pi = 1 \text{ shift}=0 \)) | \( 2L^2 w_L \) | 6    | 4L/4L                   |
| (DFS, \( \pi = L, \pi = 1/2 \text{ shift}=L/4 \)) | \( L/2 * L^2 w_L \) | 8    | 3.25L/2.75L             |
| (DFS, \( \pi = L, \pi = 1/3 \text{ shift}=L/6 \)) | \( L/3 * L^2 w_L \) | 10   | 2.83L/2.5L              |
| (DFS, \( \pi = L, \pi = 1/4 \text{ shift}=L/8 \)) | \( L/4 * L^2 w_L \) | 12   | 2.625L/2.375L           |
| (SFS/PNT, \( \pi = 0, \pi = 1, \text{shift}=0, \epsilon=3 \)) | \( (L/4)^2 w_L \) | 4    | 4L/3.75L                |
| (SFS/PNT, \( \pi = L, \pi = 1, \text{shift}=0, \epsilon=3 \)) | \( (L/4)^2 w_L \) | 4    | 4L/3.75L                |
| (SFS/PNT, \( \pi = 2L, \pi = 1, \text{shift}=0, \epsilon=3 \)) | \( (L/4)^2 w_L \) | 4    | 4L/4L                   |
| (DFS/PNT, \( \pi = 0, \pi = 1, \text{shift}=0, \epsilon=4 \)) | \( (L/5)^2 w_L \) | 4    | 4L/3.8L                 |
| (DFS/PNT, \( \pi = L, \pi = 1, \text{shift}=0, \epsilon=6 \)) | \( (L/7)^2 w_L \) | 4    | 4L/3.857L               |
| (DFS/PNT, \( \pi = L, \pi = 1, \text{shift}=L/4, \epsilon=3 \)) | \( (L/2)^2 w_L \) | 8    | 4L/3.75L                |
| (DFS/PNT, \( \pi = L, \pi = 1, \text{shift}=L/8, \epsilon=3 \)) | \( (L/4)^2 w_L \) | 8    | 4.125L/3.875L           |
| (DFS/PNT, \( \pi = L, \pi = 1, \text{shift}=L/10, \epsilon=4 \)) | \( (L/5)^2 w_L \) | 8    | 4.1L/3.9L               |
| (DFS/PNT, \( \pi = L, \pi = 1, \text{shift}=L/12, \epsilon=5 \)) | \( (L/6)^2 w_L \) | 8    | 4.083L/3.9166L          |
| (DFS/PNT, \( \pi = L, \pi = 1, \text{shift}=L/14, \epsilon=6 \)) | \( (L/7)^2 w_L \) | 8    | 4.071L/3.9286L          |

Note that, when \( \pi > 1 \), both the storage requirements and the decoding delay are increased without any reduction in the number of DFUs; therefore architectures with \( \pi > 1 \) are not suggested.
Double-flow structures lead to efficiency only for $\Pi = L$, since only for this value of $\Pi$ memory sharing can take place. For all other values of $\Pi$ the memory requirements and the number of DFUs are doubling for double-flow structures with the added advantage of doubling the MAP decoder speed since the frame decoding is split between the two flows. For (DFS, $\Pi = L$, $\pi$, shift) decoders,

storage req. [bits] $= \pi L 2^G w_L$,

number of DFUs = \begin{cases} 6, & \pi \geq 1, \\ 2 \left( 2 + \frac{1}{\pi} \right), & \pi < 1, \end{cases}

decoding delay = \begin{cases} \left( 4\pi - \frac{3}{2} \right)L or \left( 3\pi - \frac{3}{2} \right)L, & \pi > 1, \\ \left( \frac{5\pi}{2} + 2 \right)L or \left( \frac{3\pi}{2} + 2 \right)L, & \pi \leq 1, \end{cases}

shift = \begin{cases} \frac{(2\pi - 1)L}{2}, & \pi > 1, \\ \frac{\pi L}{2}, & \pi \leq 1. \end{cases}

(11)

Setting $\pi$ above 1 results in prohibitively large decoding delay. Even if $\pi$ is only 3, the decoding delay is $\left( 4\pi - \frac{3}{2} \right) L / (3\pi - 3/2) L = 10.5L/7.5L$, which is unacceptably large, therefore architectures for $\pi \leq 1$ are of interest only.

4.4. Pointer Technique-Based SISO-MAP Decoders. Figures 11 and 12 illustrate the DFG and the SM-CM for a (SFS/PNT, $\Pi = 2L$, $\pi = 1$, $\epsilon$ Pts) structure.

In all of the above structures all the necessary metrics are stored in the RAM memories. Another idea is to selectively store some of the state metrics and recompute all the others. This idea of selective recomputation leads to reduced storage requirements.

The pointers are nothing else but the actual state metric values, which are stored and extracted from a storage device whenever is necessary. The values to be stored in pointers are depicted with small white circles in Figure 11.

In this structure, during time $t = 2L t_s$ to $(3L - 1)t_s$, after $L$ dummy metric recursion steps, the obtained valid $B_l$ value is stored in a register. This value is also used to initialise a backward recursion, which operates during the symbol periods $3L$ to $4L - 1$. This backward B-metric DFU as progresses stores in registers the values $B_{l(4L/3)}$ and $B_{l(4L/2)}$ and in the RAM memory the last $L/4$ state metric values from $B_{l(L/4-1)}$ down to $B_0$. Thus, only $L/4$ state vectors are stored in RAM. An extra backward DFU will then extract the pointer $B_{l(4L/3)}$ to recalculate the other $L/4$ B-metric values ($B_{l(L/2-1)}$ down to $B_{l(L/4)}$) and store them in the memory. This procedure will be repeated until the calculation of all $L$ B-metric vectors in the window. As the backward metrics are calculated, a forward DFU computes the corresponding A-state metrics.

As can be observed from Figure 11 at any given time only $L/4$ state metric vectors of size $2G w_L$ each are stored in the RAM. This means the storage requirements of this structure are a RAM of size $(L/4) \times 2G \times w_L$ bits and 3 pointer vectors of $2G \times w_L$ bits each.

For pointer-based SFS and $\Pi = 0$ the storage of state metrics starts right after the dummy metric calculation has computed a valid metric, so there is no time space for pointers. $\Pi = 2L$ is equally well with the $\Pi = L$ case regarding pointers in the sense that the trade-offs mentioned before are not affected.

For (SFS/PNT, $\Pi = 2L$, $\pi = 1$, shift = $-\epsilon$) structures,

storage req. [bits] $= \left[ \frac{L}{\epsilon + 1} \right] 2^G w_L$,

number of DFUs = 4,

decoding delay = $4L$ or $\left( 4 - \left( \frac{\epsilon}{\epsilon + 1} \right) \right) L$.

(12a), (12b), (12c)

Increasing $\epsilon$ (the number of pointers) reduces the RAM memory but has no effect on the number of DFUs. The size of the required memory is given by $\left[ L / (\epsilon + 1) \right] 2^G w_L$. Of course, some registers need to be allocated for each pointer, whose size is $2^G w_L$. The decoding delay is only affected by the amount of pointers used if interleaver LLR reversing is employed. In this case the decoding delay is $4L - (1/\epsilon + 1)L$ and holds unless $\pi > \epsilon$.

If the structures (SFS/PNT, $\Pi$, $\pi = 1$, shift = $-\epsilon$, $\epsilon = 3$) with $\Pi = 0, L, 2L$, are compared with the corresponding SFS structures that use no pointers, it is observed that using three pointers and one more DFU results in 75% reduction in the required RAM memory. The decoder delay of the pointer-based structures when interleaver LLR reversing is employed is increased by 0.75$L$ for the first two cases $\Pi = 0, L$.

For $\pi \neq 1$, if mod($\epsilon + 1, 1/\pi$) $\neq 0$, the pointer-based techniques (single flow and double flow) are not applicable. If mod($\epsilon + 1, 1/\pi$) $= 0$, the resultant architecture does not reduce memory size but has higher throughput (computation speeds up $\pi$ times). In this case, (12a) and (12c) hold as they are, but in (12b) the number of DFUs is given by $1/\pi + 3$.

$\Pi = 2L$ is the best choice for pointer-based SFS, whereas $\Pi = L$ is a better choice for pointer-based DFS, since the latter...
di in separate memory banks, whereas in the former it cannot, parity redundant information can be separated and stored binary turbo receiver, since in the latter systematic and bank is sent to the decoder each time. This ensures that zeros are inserted at even or odd locations, and one memory frame formation block is di with the more than one diversity received symbols, the architecture using a double flow with \( \Pi \) but not the data flow unit count of the structures.

\[ \text{storage req. [bits]} = \frac{L}{(\epsilon + 1)2^Gw_L}, \]

\[ \text{number of DFUs} = 8, \]

\[ \text{decoding delay} = \left( 4 + \left( \frac{\epsilon}{2(\epsilon + 1)} \right) \right)L \]

\[ \text{or} \left( 4 - \left( \frac{\epsilon}{2(\epsilon + 1)} \right) \right)L, \]

\[ \text{shift} = \frac{L}{2(\epsilon + 1)}. \]

Again, note that increasing \( \epsilon \) reduces the storage requirement but not the data flow unit count of the structures.

Figure 13 shows a complete SISO-MAP decoder VLSI architecture using a double flow with \( \Pi = L, \pi = 1, 3 \) Pts, shift = \( L/8 \). This architecture requires 8 DFUs (2 dummy, 2 pointer saving, and 2 forward and 2 backward) and two RAM memories \( (L/8)2^Gw_L \) each and has a decoding delay of 4L. With selective recomputation only a fraction of the required state metric vectors are stored in the RAM memories. If \( \epsilon \) pointers are used, then only \( L/(\epsilon + 1) \) of the state metrics need to be stored in the RAM memory. The remaining \( L - (L/(\epsilon + 1)) \) is recalculated in blocks of \( L/(\epsilon + 1) \).

5. Conclusions

The VLSI architectures of space-time turbo trellis coding decoders as well as of a set of SISO-MAP component channel decoders used in turbo coding are proposed and investigated. The space-time turbo code receiver as opposed to binary turbo codes is based on nonbinary trellises, which imposes a number of differences. Except the fact that channel estimation as well as MRRC combining is needed to cope with the more than one diversity received symbols, the frame formation block is different than in a traditional binary turbo receiver, since in the latter systematic and parity redundant information can be separated and stored in separate memory banks, whereas in the former it cannot, so the whole symbol is stored in a memory bank. The difference is that in a STTuTC receiver case, the equalised symbols are demultiplexed and stored in memory banks, zeros are inserted at even or odd locations, and one memory bank is sent to the decoder each time. This ensures that the SISO-MAP decoder accepts the information from the corresponding encoder. The symbols are demapped by a symbol hard or soft demapper.

The proposed STTuTC architectures are based on a different ACSO unit than the binary turbo codes. This ACSO unit can handle the more than two state and transition metric pairs iteratively. In nonbinary trellises the state and transition metric pairs are more than two, because of the increased number of transitions in between states. Thus, the ACSO unit must either grow or work iteratively. Many ACSO units can be used to calculate all state metrics in one step without the need for storing those values in a state metric memory. Thus, the state metric calculation modules are considerably different than in the binary turbo decoders. The LLR calculation module is also different because more than one LLR value needs to be calculated in every decoding step. “f” ACSO units appropriately connected rather than two are required to calculate the appropriate amount of LLR values for all possible symbols. This also means that the number of soft LLR outputs the SISO-APP demapper delivers every time depends on the “f” possible values. That is, f-1 LLR must be outputted, so the LLR memory size is different than the binary turbo decoder case.

A parameter set (FS, \( \Pi, \pi, \text{shift}, \epsilon \)) helped in the comparison and led to defining equations for many different cases, single-flow, double-flow, and pointer-based techniques. Table 3 gives a list of formulae, which determine the quantities of memory size, number of deployed DFUs, and decoding delay of the most efficient techniques from those discussed in the this paper.

| Parameter set       | Mem. req. [bits] | DFUs | Decoding delay | Ptrs |
|---------------------|-----------------|------|----------------|------|
| SFS, \( \Pi = 0, \pi = 1, \) \( sft = \text{No}, \epsilon = 0 \) | \( L\pi 2^Gw_L \) | 2+1/\( \pi \) | \( (2 + 2\pi)L/(2 + \pi)L \) | 0 |
| DFS, \( \Pi = L, \pi = 1, \) \( sft = L/2, \epsilon = 0 \) | \( L\pi 2^Gw_L \) | 4+2/\( \pi \) | \( 2L + 5sft/2L + 3sft \) | 0 |
| SFS/PNT, \( \Pi = 0, \epsilon = 1, \) \( sft = \text{No}, \epsilon \) | \((L/(\epsilon + 1))2^Gw_L \) | 4 | \( 4L/4L - (1/(\epsilon + 1))L \) | \( \epsilon 2^Gw_L \) |

### References

[1] C. Berrou, A. Glavieux, and P. Thitimajshima, “Near Shannon limit error-correcting coding and encoding: turbo-codes,” in *Proceedings of the IEEE International Conference on Communications (ICC '93)*, vol. 2, pp. 1064–1070, Geneva, Switzerland, May 1993.

[2] C. Berrou and A. Glavieux, “Near optimum error correcting coding and decoding: turbo codes,” *IEEE Transactions on Communications*, vol. 44, no. 10, pp. 1261–1271, 1996.

[3] J. Hagenauer and P. Hoher, “A Viterbi Algorithm with soft-decision outputs and its applications,” in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '89)*, vol. 3, pp. 1680–1686, 1989.

[4] C. Berrou, P. Adde, E. Angui, and S. Faudeil, “Low complexity soft-output viterbi decoder architecture,” in *Proceedings of the IEEE International Conference on Communications (ICC '93)*, pp. 737–740, May 1993.
[5] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, “Optimal decoding of linear codes for minimizing symbol error rate,” *IEEE Transactions on Information Theory*, vol. 20, no. 2, pp. 284–287, 1974.

[6] D. Tujkovic, “Recursive space-time trellis codes for turbo coded modulation,” in Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM ’01), San Francisco, Calif, USA, 2000.

[7] D. Cui and A. M. Haimovich, “Performance of parallel concatenated space-time codes,” *IEEE Communications Letters*, vol. 5, no. 6, pp. 236–238, 2001.

[8] W. Firmanto, B. Vucetic, J. Yuan, and Z. Chen, “Space-time turbo trellis coded modulation for wireless data communications,” *Eurasip Journal on Applied Signal Processing*, vol. 2002, no. 5, pp. 459–470, 2002.

[9] Y. Hong, J. Yuan, Z. Chen, and B. Vucetic, “Space-time turbo trellis codes for two, three, and four transmit antennas,” *IEEE Transactions on Vehicular Technology*, vol. 53, no. 2, pp. 318–328, 2004.

[10] W. Firmanto, Z. Chen, B. Vucetic, and J. Yuan, “Design of space-time turbo trellis coded modulation for fading channels,” in *IEEE Global Telecommunications Conference (GLOBECOM’01)*, pp. 1093–1097, November 2001.

[11] G. Masera, G. Piccinini, M. R. Roch, and M. Zamboni, “VLSI architectures for turbo codes,” *IEEE Transactions on Very Large Scale Integration Systems*, vol. 7, no. 3, pp. 369–379, 1999.

[12] Y. Tong, T. H. Yeap, and J. Y. Chouinard, “VHDL implementation of a turbo decoder with log-MAP-based iterative decoding,” *IEEE Transactions on Instrumentation and Measurement*, vol. 53, no. 4, pp. 1268–1278, 2004.

[13] C. M. Wu, M. D. Shieh, C. H. Wu, Y. T. Hwang, and J. H. Chen, “VLSI architectural design tradeoffs for sliding-window Log-MAP decoders,” *IEEE Transactions on Very Large Scale Integration Systems*, vol. 13, no. 4, pp. 439–447, 2005.

[14] C. Chaikalis and J. Noras, “Reconfigurable turbo decoding for 3G applications,” *Signal Processing*, vol. 84, no. 10, pp. 1957–1972, 2004.

[15] C. Chaikalis, “Implementation of a reconfigurable turbo decoder in 3GPP for flat rayleigh fading,” *Digital Signal Processing*, vol. 18, no. 2, pp. 189–208, 2008.

[16] L. Yonghui, B. Vucetic, Z. Qishan, and H. Yi, “Assembled space-time turbo trellis codes,” *IEEE Transactions on Vehicular Technology*, vol. 54, no. 5, pp. 1768–1772, 2005.

[17] M. M. Mansour and N. R. Shanbhag, “VLSI Architectures for SISO-APP Decoders,” *IEEE Transactions on Very Large Scale Integration Systems*, vol. 11, no. 4, pp. 627–650, 2003.

[18] E. Boutillon, W. J. Gross, and P. G. Gulak, “VLSI architectures for the MAP algorithm,” *IEEE Transactions on Communications*, vol. 51, no. 2, pp. 175–185, 2003.

[19] K. C. Nguyen, U. Gunawardana, and R. Liyana-Pathirana, “Optimization of sliding window algorithm for space-time turbo trellis codes,” in *Proceedings of the International Symposium on Communications and Information Technologies (ISCIIT ’07)*, pp. 1182–1186, October 2007.

[20] A. Worm, H. Lamm, and N. Wehn, “VLSI architectures for high-speed MAP decoders,” in *Proceedings of the 14th International Conference on VLSI Design*, pp. 446–453, January 2001.

[21] C. Schurgers, F. Cathoor, and M. Engels, “Memory optimization of MAP turbo decoder algorithms,” *IEEE Transactions on Very Large Scale Integration Systems*, vol. 9, no. 2, pp. 305–312, 2001.

[22] S. S. Pietrobon, “Implementation and performance of a turbo/MAP decoder,” *International Journal of Satellite Communications*, vol. 16, no. 1, pp. 23–46, 1998.

[23] A. Worm, H. Michel, F. Gilbert, G. Kreiselmaier, M. Thul, and N. Wehn, “Advanced implementation issues of turbo decoders,” in *Proceedings of the 2nd International Symposium on Turbo Codes*, pp. 351–354, Brest, France, September 2000.

[24] S. ten Brink, “Convergence of iterative decoding,” *Electronics Letters*, vol. 35, no. 13, pp. 1117–1118, 1999.

[25] D. Divsalar, S. Dolinar, and F. Pollara, “Iterative turbo decoder analysis based on density evolution,” *IEEE Journal on Selected Areas in Communications*, vol. 19, no. 5, pp. 891–907, 2001.

[26] S. ten Brink, “Convergence behavior of iteratively decoded parallel concatenated codes,” *IEEE Transactions on Communications*, vol. 49, no. 10, pp. 1727–1737, 2001.
Research Article

Interference Cancellation Using Replica Signal for HTRCI-MIMO/OFDM in Time-Variant Large Delay Spread Longer Than Guard Interval

Yuta Ida,1 Chang-Jun Ahn,2 Takeshi Kamio,1 Hisato Fujisaka,1 and Kazuhisa Haeiwa1

1 Graduate School of Information Sciences, Hiroshima City University, 3-4-1 Ozukahigashi, Asaminami-ku, Hiroshima 731-3194, Japan
2 Graduate School of Engineering, Chiba University, 1-33 Yayoi-cho, Inage-ku, Chiba 263-8522, Japan

Correspondence should be addressed to Yuta Ida, y.ida@chiba-u.jp

Received 15 July 2011; Revised 22 February 2012; Accepted 12 March 2012

Academic Editor: Hanho Lee

Copyright © 2012 Yuta Ida et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Orthogonal frequency division multiplexing (OFDM) and multiple-input multiple-output (MIMO) are generally known as the effective techniques for high data rate services. In MIMO/OFDM systems, the channel estimation (CE) is very important to obtain an accurate channel state information (CSI). However, since the orthogonal pilot-based CE requires the large number of pilot symbols, the total transmission rate is degraded. To mitigate this problem, a high time resolution carrier interferometry (HTRCI) for MIMO/OFDM has been proposed. In wireless communication systems, if the maximum delay spread is longer than the guard interval (GI), the system performance is significantly degraded due to the intersymbol interference (ISI) and intercarrier interference (ICI). However, the conventional HTRCI-MIMO/OFDM does not consider the case with the time-variant large delay spread longer than the GI. In this paper, we propose the ISI and ICI compensation methods for a HTRCI-MIMO/OFDM in the time-variant large delay spread longer than the GI.

1. Introduction

Recently, the sophisticated terminal as a smart phone becomes widely used and many multimedia services are provided [1, 2]. High speed packet access (HSPA) using a wideband code division multiplexing access (W-CDMA) is used in the mobile communications and provides the data services with the maximum transmission rate of about 14 Mbps [3]. However, since the available frequency band is limited, a HSPA cannot improve the transmission rate more than now [4]. To solve this problem, orthogonal frequency division multiplexing (OFDM) and multiple-input multiple-output (MIMO) are actively used [5–9]. OFDM is a multicarrier digital modulation. The OFDM signal can be transmitted in parallel by using the many subcarriers that are mutually orthogonal. In MIMO systems, the signal of the several transmit antennas is transmitted in the same frequency band. Moreover, since each transmitted signal is sent over the independent channel, the space diversity can be obtained in the receiver. Therefore, a long term evolution (LTE) has been standardized as 3.9 G system using MIMO/OFDM systems [10, 11]. An LTE provides the maximum transmission rate of about 50–100 Mbps. Moreover, an LTE Advanced will support broadband data services with the maximum transmission rate of about 100 M–1 Gbps as 4 G systems.

Since the received signal is changed due to the amplitude and phase variations for a frequency selective fading, the channel estimation (CE) is important to compensate the channel variance. In the conventional MIMO/OFDM, the orthogonal pilot symbols-based CE is used to identify an accurate channel state information (CSI) [7]. However, since the orthogonal pilot-based CE requires the large number of pilot symbols and large transmission power, the transmission rate is degraded. To mitigate these problems, a high time resolution carrier interferometry (HTRCI) for MIMO/OFDM has been proposed [12].
In wireless communications, intersymbol interference (ISI) and intercarrier interference (ICI) are serious problems to mitigate the system performance. To prevent these problems, a guard interval (GI) is generally inserted. In general, GI is usually designed to be longer than the delay spread of the channel. However, if the maximum delay spread is longer than the GI, the system performance is significantly degraded due to the ISI and ICI. However, the conventional HTRCI-MIMO/OFDM does not consider the case with the time-variant large delay spread longer than the GI. Until this time, several schemes have proposed the ISI and ICI compensation methods due to the large delay spread channel. For example, [13] has proposed the ISI reduction method by extending the GI. However, since [13] has extended the GI length, the maximum throughput is degraded and the transmission power is also increased. Reference [14] has proposed the ISI and ICI compensation methods using the turbo equalizer. However, [14] has large complexity by the iterative processing for the turbo equalizer. Reference [15] has proposed the ISI and ICI compensation methods using the estimated channel coefficients and the CSI to reproduce the interference components. However, the packet length becomes longer by using the training symbol. To mitigate the above-mentioned problems, we propose the time domain ISI compensation method with the replica signal based on the ICI compensation for a HTRCI-MIMO/OFDM in this paper. This paper is organized as follows. In Section 2, we present the system model. Then, we describe the proposed system in Section 3. In Section 4, we show the computer simulation results. Finally, the conclusion is given in Section 5.

2. System Model

This section describes the system model, which employs the time-division multiplexing (TDM) transmission for multiple users. This system is illustrated in Figure 1.

2.1. Channel Model. We assume that a propagation channel consists of \( L \) discrete paths with different time delays. The impulse response between the \( m \)th transmit and \( n \)th receive antenna \( h_{m,n}(\tau, t) \) is represented as follows:

\[
h_{m,n}(\tau, t) = \sum_{l=0}^{L-1} h_{m,n,l}(t) \delta(\tau - \tau_{m,n,l}),
\]

where \( h_{m,n,l} \) and \( \tau_{m,n,l} \) are the complex channel gain and the time delay of the \( l \)th propagation path, and \( \sum_{l=0}^{L-1} E|h_{m,n,l}|^2 = 1 \), where \( E| \cdot | \) denotes the ensemble average operation. The channel transfer function \( H_{m,n}(f, t) \) is the Fourier transform of \( h_{m,n}(\tau, t) \) and is given by

\[
H_{m,n}(f, t) = \int_{0}^{\infty} h_{m,n}(\tau, t) \exp(-j2\pi f \tau) d\tau
= \sum_{l=0}^{L-1} h_{m,n,l}(t) \exp(-j2\pi f \tau_{m,n,l}).
\]

2.2. HTRCI-MIMO/OFDM. The transmission block diagram of the proposed system is shown in Figure 1(a). Firstly, the coded binary information data sequence is modulated, and \( N_p \) pilot symbols are appended at the beginning of the sequence. The HTRCI-MIMO/OFDM transmitted signal for
the $m$th transmit antenna can be expressed in its equivalent baseband representation as follows:

\[
    s_m(t) = \sum_{i=0}^{N_p+N_d-1} g(t-iT) \cdot \left\{ \sqrt{\frac{2\sigma}{N_c}} \sum_{k=0}^{N_s-1} u_m(k,i) \cdot \exp\left[ j2\pi(t-iT)k/T_s \right] \right\},
\]

where $N_d$ and $N_p$ are the number of data and pilot symbols, $N_c$ is the number of subcarriers, $T_s$ is the effective symbol length, $S$ is the average transmission power, and $T$ is the OFDM symbol length, respectively. The frequency separation between adjacent orthogonal subcarriers is $1/T_s$ and can be expressed by using the $k$th subcarrier of the $i$th modulation symbol $s_m(k,i)$ with $|d_m(k,i)| = 1$ for $N_p \leq i \leq N_p+N_d-1$ as follows:

\[
    u_m(k,i) = c_{\text{PN}}(k) \cdot d_m(k,i),
\]

where $c_{\text{PN}}$ is a long pseudonoise (PN) sequence as a scrambling code to reduce the peak-to-average power ratio (PAPR). GI is inserted in order to eliminate the ISI due to a multipath fading, and hence, we have

\[
    T = T_g + T_s,
\]

where $T_g$ is the GI length. In communication systems, $\varepsilon$ is generally considered as 4 or 5, where $\varepsilon = T_s/T_g$. In this paper, we assume $\varepsilon = 4$. In (3), the transmission pulse $g(t)$ is given by

\[
    g(t) = \begin{cases} 1 & \text{for } -T_g \leq t \leq T_s \\ 0 & \text{otherwise}. \end{cases}
\]

For $0 \leq i \leq N_p - 1$, the transmitted pilot signal of the $k$th subcarrier for the $m$th transmit antenna element is given by

\[
    d_m(k,i) = \sum_{n=0}^{N_s-1} \exp\left[ -j2\pi(j\overline{m} + 2n)T_gk/T_s \right] \quad \text{for } i = \left\lfloor \frac{m}{\zeta} \right\rfloor
\]

\[
    = 0 \quad \text{otherwise},
\]

where $\zeta = \lceil \varepsilon/2 \rceil$, $\overline{m} = \operatorname{mod}(m,\zeta)$, and $\lfloor x \rfloor$ stands for the integer lower and closer to $x$, respectively. In this case, the HTRCI-MIMO/OFDM pilot signal can multiplex the same impulse responses from the transmit antenna elements in each receive antenna element in $\zeta$ times on the time domain without overlapping to each other. For example, if we consider $\varepsilon = 4$ and 2 transmit antenna elements, we obtain $\sum_{k=0}^{N_s-1} d_0(k,0) = \{1,0,\ldots,1,0\}$ and $\sum_{k=0}^{N_s-1} d_1(k,0) = \{1,0,-1,0,\ldots,1,0,-1,0\}$ as the pilot signal of the first and second transmit antenna elements from (7) as shown in Figure 2(a). In this case, since each pilot signal contains "0" components, we can identify the half of transmission power compared with the conventional MIMO/OFDM system using the orthogonal pilot. However, if $\tau_{m,n,L} > T_g$, the conventional HTRCI-MIMO/OFDM cannot obtain an accurate CSI, where $\tau_{m,n,L}$ is the maximum delay spread for the $m$th transmit and $n$th receive antenna. To solve this problem, we propose the new channel estimation using a HTRCI-MIMO/OFDM. Firstly, in the transmitter, we shift the channel impulse responses of (7) as follows:

\[
    d_m(k,i) = \begin{cases} \sum_{n=0}^{N_s-1} \exp\left[ -j2\pi(j\overline{m} + 2n)T_gk/T_s \right] \quad \text{for } i = \left\lfloor \frac{m}{\zeta} \right\rfloor \\ 0 \quad \text{otherwise}. \end{cases}
\]

In (8), the channel impulse responses are shifted as shown in Figure 2(a). This operation enables estimating the maximum delay spread longer than the GI and to obtain an accurate CSI in the receiver.

The received structure is illustrated in Figure 1(b). By applying the FFT operation, the received signal $r_n(t)$ is resolved into $N_r$ subcarriers. The received signal for the $n$th receive antenna $r_n(t)$ in the equivalent baseband representation can be expressed as follows:

\[
    r_n(t) = \frac{M-1}{T_s} \int_{-\infty}^{\infty} h_{m,n}(\tau,t)s_m(t-\tau)d\tau + n_n(t),
\]

where $M$ is the number of transmit antennas and $n_n(t)$ is additive white Gaussian noise (AWGN) with a single-sided power spectral density of $N_0$ for the $n$th receive antenna, respectively. The $k$th subcarrier $\hat{n}_n(k,i)$ is given by

\[
    \hat{n}_n(k,i) = \left\{ \begin{array}{ll}
    1 & \text{for } -T_g \leq t \leq T_s \\
    0 & \text{otherwise}
    \end{array} \right.
\]

\[
    \cdot \left\{ \sum_{n=0}^{N_s-1} \exp\left[ -j2\pi(j\overline{m} + 2n)T_gk/T_s \right] \times h_{m,n}(\tau,t+iT)g(t-\tau) \right\} dt + \hat{n}_n(k,i),
\]

where $\hat{n}_n(k,i)$ is AWGN noise with zero mean and a variance of $2N_0/T_s$. After abbreviating, (10) can be rewritten as follows:

\[
    \hat{n}_n(k,i) = \frac{1}{T_s} \int_{-\infty}^{\infty} h_{m,n}(\tau,t+iT)g(t-\tau) \cdot \left\{ \sum_{n=0}^{N_s-1} \exp\left[ -j2\pi(j\overline{m} + 2n)T_gk/T_s \right] \right\} dt + \hat{n}_n(k,i),
\]
After descrambling, the output signal \( \hat{r}_n(k, i) \) for the \( n \)th receive antenna element is given by

\[
\hat{r}_n(k, i) = \frac{c_{PN}^*(k)}{|c_{PN}(k)|^2} \hat{r}_n(k, i) \\
= \sqrt{\frac{2S}{N_c}} \sum_{m=0}^{M-1} T_s H_{m,n} \left( \frac{k}{T_s}, iT \right) d_m(k, i) + \tilde{n}_n(k, i),
\]

(12)

where \((\cdot)^*\) is a complex conjugate and \(\frac{c_{PN}^*(k)}{|c_{PN}(k)|^2}\) is the descrambling operation, respectively. Observing (11) and (12), the noise components are the same notation. In this paper, the descrambling operation is to rotate the phase of each subcarrier by using a PN code. Since \(\varepsilon = T_d/T_g\), a HTRCI-MIMO/OFDM can multiplex the same impulse responses in \(\zeta\) times on the time domain. After the pilot signal separation, the pilot signal is converted to the time domain signal \(\hat{r}_n(t)\) again as

\[
\hat{r}_n(t) = \sum_{i=0}^{N_i-1} \sqrt{2P} \sum_{k=0}^{N_c-1} \hat{r}_n(k, i) \exp \left[ \frac{j2\pi(t - iT)k}{T_s} \right] \\
= \sum_{i=0}^{N_i-1} \sqrt{2P} \sum_{m=0}^{M-1} \sum_{k=0}^{N_c-1} h_{m,n}(t, t + iT) d_m(k, i) \\
\cdot \exp \left[ \frac{j2\pi(t - iT)k}{T_s} \right] + \tilde{n}_n(t),
\]

(13)

where \(P\) is the transmission pilot signal power and \(\tilde{n}_n(t)\) is the noise component, respectively. Here, if \(\tau_{m,n,d} > T_g\), since
the channel impulse responses of the different time window overlap the desired channel impulse responses, the accurate CSI cannot be obtained. Therefore, we process as follows. At the receiver, in Figure 2(b), the overlapped channel impulse responses show the same amplitude for the transmitted pilot signal without the shift in the different time window. In this case, we can not estimate the maximum delay spread. On the other hand, the overlapped channel impulse responses show the different amplitude for the transmitted pilot signal with the shift in the different time window. Therefore, we can estimate the maximum delay spread from the different amplitude in the different time windows. Next, we eliminate the channel impulse responses of the different time window. The channel impulse responses of the different time window can be eliminated by using the channel impulse responses of the GI. These channel impulse responses do not contain the channel impulse responses of the different time window. This is because the pilot signal of the GI is the head of packet and it does not contain the channel impulse responses of the different time window. Therefore, we can eliminate the channel impulse responses of the different time window. From the above process, the frequency response of the kth subcarrier between the mth transmit and nth receive antenna, \( \hat{H}_{m,n}(k) \) for \( \tau_{m,n,L} > T_g \) is obtained by

\[
\hat{H}_{m,n}(k) = \sqrt{\frac{N_c}{2T}} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \frac{1}{T_s} \cdot \int_0^{T_s} \left\{ \sum_{l=0}^{L-1} \sum_{\ell=0}^{L-1} h_{m,n,l} (t + \rho T) \cdot \int_{-\infty}^{\infty} \left[ \sum_{\mu=0}^{L-1} \delta (t - \tau_{m,n,l} - (2\mu + \rho)T_s) \right. \\
\left. - \delta (t - \tau_{m,n,l}) \right] \cdot \exp \left( -\frac{j2\pi \nu t}{T_s} \right) dt \right\} dt + \eta_{m,n}(k) \text{ for } L' < T_g, \rho = \left[ \frac{m}{T_s} \right],
\]

where \( \eta_{m,n}(k) \) is AWGN component with \( E[|\eta_{m,n}(k)|]^2 = E[|\hat{h}_{m}(k,i)/\xi|^2] = \sigma^2/\xi \).

3. Proposed System

3.1. Rewritten Matrix Form. After the pilot signal separation, (13) for \( \tau_{m,n,L} \leq T_g \) can be rewritten in the matrix form as follows:

\[
R_{i,n} = \sum_{m=0}^{M-1} \lambda_{i,m,n} \mathbf{F} \mathbf{D}_{i,m} + N_{i,n},
\]

where \( \lambda_{i,m,n} \) is the \( N_c \times N_f \) time-domain channel matrix for the ith symbol between the mth transmit and nth receive antenna, \( N_{i,n} \) is the \( N_c \times 1 \) noise matrix, and \( \mathbf{F} \) is the IFFT operation, respectively. However, (15) for \( \tau_{m,n,L} > T_g \) is rewritten as follows:

\[
R_{i,n} = \sum_{m=0}^{M-1} (\lambda_{i,i-1,m,n} \mathbf{F} \mathbf{D}_{i-1,m} + \lambda_{i,i,m,n} \mathbf{F} \mathbf{D}_{i,m}) + N_{i,n},
\]

where \( \lambda_{i,i-1,m,n} \) \( \lambda_{i,i,m,n} \) denote the ISI and ICI channel matrices for the \( (i - 1) \)th and ith symbols between the mth transmit and nth receive antenna, respectively. Observing (16), since the received signal \( R_{i,n} \) contains the ISI and ICI terms, these equalization processing are necessary.

3.2. ISI and ICI Equalization. In \( \tau_{m,n,L} > T_g \), since the first data symbol of \( R_{i,n} \) does not contain the ISI [16], (16) for \( i = 0 \) is obtained as

\[
R_{0,n} = \sum_{m=0}^{M-1} \lambda_{i,i,0,m,n} \mathbf{F} \mathbf{D}_{0,m} + N_{0,n}.
\]

However, the data symbols for \( i > 0 \) have the ISI. Here, the ISI equalization is performed by using the previous detected symbol \( \mathbf{D}_{i-1,m,n} \) and the estimated ISI channel matrix \( \hat{\lambda}_{i,i,m,n} \). The estimated ISI channel matrix \( \hat{\lambda}_{i,i,m,n} \) consists of the estimated channel impulse response. The estimated channel impulse response \( \hat{h}_{m,n,f}(\tau) \) is obtained by the channel response \( \hat{H}_{m,n}(k) \) after the IFFT operation, where \( L' \leq L' < L - 1 \). Therefore, the ISI equalized signal \( \hat{R}_{i,n} \) is given by

\[
\hat{R}_{i,n} = R_{i,n} - \sum_{m=0}^{M-1} \hat{\lambda}_{i,i,m,n} \mathbf{F} \mathbf{D}_{i-1,m} + \hat{\eta}_{i,n},
\]

where \( \hat{\eta}_{i,n} \) is the noise term with the residual ISI. After the FFT operation, by using maximum likelihood detection (MLD), the detected signal \( \hat{\mathbf{D}}_i = [\hat{\mathbf{D}}_{i,0}, \ldots, \hat{\mathbf{D}}_{i,m}, \ldots, \hat{\mathbf{D}}_{i,M-1}]^T \) is obtained as follows:

\[
\hat{\mathbf{D}}_i = \arg\min_{\mathbf{D}} \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} \left\| F^{-1} \hat{R}_{i,n} - \hat{H}_{m,n} \mathbf{D}_{i,m} \right\|^2,
\]

where \((\cdot)^T\) is the transpose operation, \( N \) is the number of receive antennas, \( \hat{\mathbf{D}}_{i,m} \) is the constellation of the symbol replica candidates as \( \sum_{m=0}^{M-1} \sum_{c=0}^{N_c-1} \mathbf{D}_{i,m,c} \), \( c \) is the modulation level, and \( \hat{H}_{m,n} \) is the matrix form of \( \hat{H}_{m,n}(k) \), respectively. From (19), ISI is eliminated from the received signal. However, the orthogonality is destroyed by the detected signal due to the ISI compensation. If ICI is eliminated from (19), the detected signal \( \hat{\mathbf{D}}_{i,m} \) can be more accurately detected.

3.3. Replica Signal Insertion Based on ICI Equalization. To eliminate the ICI, we consider the orthogonality reconstruction with inserting the detected signal after the ISI
compensation. The ICI equalized signal $\hat{R}_{i,n}$ with inserting eliminated the part of signal using the previous detected symbol $\hat{D}_{i,m}$ is given by

$$
\hat{R}_{i,n} = \hat{R}_{i,n} + \sum_{m=0}^{M-1} \lambda_{ic,n,m} F \hat{D}_{i,m} = \sum_{m=0}^{M-1} \lambda_{ic,n,m} F \hat{D}_{i,m} + N_{i,n},
$$

(20)

where $\lambda_{ic,n,m}$ is the estimated ICI channel matrix and $N_{i,n}$ is the noise term with the residual ISI and residual ICI, respectively. $\lambda_{ic,n,m}$ consists of the estimated channel impulse responses $\hat{h}_{m,n}(t) \delta(t - \tau_{m,n})$. After the FFT operation, the detected signal $\hat{D}_i = [\hat{D}_{i,0}, \ldots, \hat{D}_{i,m}, \ldots, \hat{D}_{i,M-1}]^T$ is obtained as follows:

$$
\hat{D}_i = \arg \min_{\hat{D}} \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} \left| F^{-1} \hat{R}_{i,n} - \hat{h}_{m,n} \hat{D}_{i,m} \right|^2.
$$

(21)

Observing (19) and (21), since ICI is eliminated, $\hat{D}_{i,m}$ is the more accurately detected signal compared with $\hat{D}_{i,m}$.

3.4. Complexity Comparison. Here, we compare the complexity of the conventional and proposed methods. Firstly, we show the complexity of [14]. The authors of [14] have proposed the ISI and ICI compensations using the turbo equalization. From Table 1 of [14], the complexity of [14] proposed the ISI and ICI compensations using the turbo equalization. From Table 1 of [14], the complexity of [14] is given by

$$
C_{[14]} = \{NNcMNc + 4NcNM^2 + 3NcNM
$$

$$
+ NNc \log_2 Nc + 6N(Nc + L)MNc\}N_d
$$

(22)

For (23), the first term is the complexity of MLD and the second term is the complexity of IFFT. Next, the complexity of (20) is obtained by

$$
C_{c} = NNc(N_d - 1)CM + \{MNc(N_d - 1)\} \log Nc.
$$

(23)

For (24), and (25) are the complexity of MLD and IFFT of (23). Finally, the complexity of (21) is

$$
C_{MLD} = NNcN_dCM = \{NNc(N_d - 1) + NNc\}CM.
$$

(24)

Therefore, the complexity of the proposed method is obtained by

$$
C_{pro} = C_{c} + C_{ici} + C_{MLD}
$$

$$
= Nc\left\{N(3N_d - 1)CM + M(N_d - 1) \log Nc\right\}.
$$

(25)

For example, when $M = N = 2, Nc = 64, N_d = 20, N_t = 80, L = 2, and C = 4$, the complexities of [14] and the proposed method are 3000320 and 130560 from (22) and (26). Therefore, the proposed method is small compared with [14].

4. Computer Simulated Results

In this section, we show the performance of the proposed method. Figure 1 shows the simulation model of the proposed system. In this simulation, we assumed that $2 \times 2$ and $4 \times 4$ MIMO systems. On the transmitter, the pilot signal is assigned for each transmitter using (8). In this case, the proposed system can multiplex the same impulse responses from the transmit antenna elements in each receive antenna element in $\zeta$ times on the time domain. These have been found to be efficient for the transmission of the OFDM signal over the frequency selective fading channel. After serial to parallel (S/P) converted, the coded bits are QPSK modulated, and then the pilot signal and data signal are multiplexed with the scrambling using a PN code to reduce the PAPR. The OFDM time signal is generated by the IFFT operation and is transmitted to the frequency-selective and time-variant radio channel after the cyclic extension has been inserted. The transmitted signal is subject to the broadband channel propagation. In this simulation, we assume that OFDM symbol period is $5 \mu s$ and guard interval is $1 \mu s$ for $\epsilon = 4$, and $L = 2, 3$ path Rayleigh fading. The maximum Doppler frequency is $100$ Hz. In the receiver, guard interval is erased from the received signal and the received signal is converted S/P. The parallel sequences are passed to the FFT operator and convert the signal back to the frequency domain. After the descrambling and IFFT operation, each impulse response for all combination of the transmit and receive antenna elements can be estimated by extracting and averages $\zeta$ times impulse responses using the time windows with (14). In $T_{m,n,L} > T_g$, the HTRCI-MIMO/OFDM pilot signal contains the overlapped channel impulse responses from the different time window. However, the pilot signal of the GI does not contain the overlapped channel impulse responses. By using this signal, the channel impulse responses of the different time window are eliminated from the overlapped channel impulse responses as shown in Figure 2(b). The frequency domain data signal is detected and demodulated by using the MLD algorithm. Since the detected data signal contains the ISI and ICI, these equalization processing are necessary. The ISI equalization is performed with the previous detected symbol and estimated ISI channel matrix as (18). From (18), ISI is eliminated. However, the orthogonality is destroyed by the detected signal due to the ISI compensation. To reconstruct the orthogonality, the ICI equalization is performed by using the replica signal insertion as (20). Finally, the data
signal is detected as (21). The packet consists of $N_p = 1, 2$ pilot symbols and $N_d = 20$ data symbols. Table 1 shows the simulation parameters.

Figures 3 and 4 show the BER of the conventional and proposed methods for $2 \times 2$ and $4 \times 4$ MIMO systems at Doppler frequency of 10 Hz. The number of maximum delay spread is 4 and 16 in $2 \times 2$ and $4 \times 4$ MIMO systems, respectively. From the simulation results, the BER performance for no ISI and ICI compensation is increased about 22 and 300 times compared with no ISI and ICI case. Therefore, the proposed method can eliminate the ISI and ICI. Next, we compare the 2- and 3-path models. The 2-path model contains the channel impulse responses of the different time window in the time domain HTRCI-MIMO/OFDM pilot signal. However, they do not overlap the desired channel impulse responses. On the other hand, the 3-path model overlaps the desired channel impulse responses. Therefore, the channel impulse responses of the different time window are eliminated by using the channel impulse responses of the GI. From the simulation results, the 2-path model shows the better BER performance than that of the 3-path model in low $E_b/N_0$. It means that CSI degrades due to the eliminated processing of the overlapped channel impulse responses in the 3-path model.

Figures 5 and 6 show the BER of the proposed method for $2 \times 2$ and $4 \times 4$ MIMO systems at Doppler frequency of 10 Hz. The number of maximum delay spread is 4 and 16 in $2 \times 2$ and $4 \times 4$ MIMO systems, respectively. For the proposed method without the noise, the noise is not added in the channel impulse responses of the GI. From the simulation results, the 2- and 3-path models without the noise show the approximately same BER performance. Therefore, the 3-path model with the noise degrades due to the noise of the GI.

Figures 7 and 8 show the BER versus the number of maximum delay spread for the conventional and proposed methods with $2 \times 2$ and $4 \times 4$ MIMO systems at Doppler frequency of 10 Hz. Here, $E_b/N_0$ per received antenna is 20 and 15 dBs in $2 \times 2$ and $4 \times 4$ MIMO systems, respectively. In $2 \times 2$ MIMO system, the BER performances of no ISI and ICI compensation and conventional method are increased

| Table 1: Simulation parameters. |
|--------------------------------|
| Data modulation | QPSK |
| Data detection | MLD |
| Symbol duration | 5μs |
| Frame size | $N_p = 1, 2, N_d = 20$ |
| FFT size | 64 |
| Number of carriers | 64 |
| Guard interval | 16 sample times |
| Fading | 2, 3 path Rayleigh fading |
| Doppler frequency | 10 Hz |
| Antennas | $(M, N) = (2, 2), (4, 4)$ |
about 5 times. On the other hand, the BER performance of the proposed method is increased about 2 times. Therefore, the proposed method can suppress the ISI and ICI due to the changing of the number of maximum delay spread. In 4 × 4 MIMO system, the BER performance of no ISI and ICI compensation and the conventional method are increased about 170 and 80 times. On the other hand, the BER performance of the proposed method is increased about 4 times and the proposed method can suppress the ISI and ICI as 2 × 2 MIMO system. However, the BER performance

---

**Figure 5:** The BER of the proposed method for 2 × 2 MIMO system at Doppler frequency of 10 Hz.

**Figure 6:** The BER of the proposed method for 4 × 4 MIMO system at Doppler frequency of 10 Hz.

**Figure 7:** The BER versus the number of maximum delay spread for the conventional and proposed methods with 2 × 2 MIMO system at Doppler frequency of 10 Hz.

**Figure 8:** The BER versus the number of maximum delay spread for the conventional and proposed methods with 4 × 4 MIMO system at Doppler frequency of 10 Hz.
of the proposed method with the 2-path model is increased about 36 times compared with 3-path model. This is because the channel impulse responses of the GI for the second time domain HTRCI-MIMO/OFDM pilot symbol are overlapped from the first pilot symbol in 4 MIMO system.

Figure 9 shows the throughput performance of the conventional and proposed methods for 2 × 2 and 4 × 4 MIMO systems at Doppler frequency of 10 Hz.

\[
T_p = \frac{N_d \cdot N_c \cdot C \cdot R \cdot M}{(N_p + N_d)T} \cdot (1 - P_{\text{per}}),
\]

(27)

where R is the coding rate and \(P_{\text{per}}\) is the packet error rate (PER), respectively. The extension of the GI length is the simple method to prevent the problem of the maximum delay spread [13]. However, the throughput performance is degrade with this method. This is because the number of carriers \(N_c\) decreases for (27) when the OFDM symbol length \(T\) is the same as no ISI and ICI case. In this simulation, \(N_c\) decreases from 64 to 48, and the GI length \(T_g\) increases from 16 to 32. On the other hand, the proposed method can maintain the number of carriers \(N_c\). From (27), the maximum throughput for no ISI and ICI case is about 48.8 and 93.1 Mbps in 2 × 2 and 4 × 4 MIMO systems, respectively. Therefore, the proposed method shows the best throughput performance and can achieve the maximum throughput performance as the same no ISI and ICI case.

5. Conclusion

In this paper, we have focused on the large delay spread channel and proposed the ISI and ICI compensation methods for a HTRCI-MIMO/OFDM. In the proposed method, we have performed the time domain ISI compensation method with the replica signal based on the ICI compensation. From the simulation results, the proposed method has achieved the approximately same BER performance like the case with no ISI and ICI. Moreover, the proposed method can suppress the ISI and ICI due to the changing of the number of maximum delay spread. Finally, the proposed method has shown the best throughput performance and can achieve the same maximum throughput performance compared with no ISI and ICI case.

References

[1] “2010 White paper information and communications in Japan,” http://www.soumu.go.jp/johotsusintokei/whitepaper/ja/h22/pdf/index.html.
[2] H. Shoki and Y. Tanabe, “How fast will be required, how fast can be realized in future wireless communications?” IEICE B+ Pluse, no. 11, pp. 12–22, 2009.
[3] F. Adachi, H. Tomeba, and K. Takeda, “Introduction of frequency-domain signal processing to broadband single-carrier transmissions in a wireless channel,” IEICE Transactions on Communications, vol. E92-B, no. 9, pp. 2789–2808, 2009.
[4] S. Abeta, “Toward LTE commercial launch and future plan for LTE enhancements (LTE-Advanced),” in 12th IEEE International Conference on Communication Systems (ICCS ’10), pp. 146–150, November 2010.
[5] L. J. Cimini, “Analysis and simulation of a digital mobile channel using orthogonal frequency division multiplexing,” IEEE Transactions on Communications, vol. 33, no. 7, pp. 665–675, 1985.
[6] J. A. C. Bingham, “Multicarrier modulation for data transmission: an idea whose time has come,” IEEE Communications Magazine, vol. 28, no. 5, pp. 5–14, 1990.
[7] A. van Zelst, R. van Nee, and G. A. Water, “Space division multiplexing (SDM) for OFDM systems,” in 51st Vehicular Technology Conference (VTC ’00), pp. 1070–1074, May 2000.
[8] I. Koffman and V. Roman, “Broadband wireless access solutions based on OFDM access in IEEE 802.16,” IEEE Communications Magazine, vol. 40, no. 4, pp. 96–103, 2002.
[9] Y. Ogawa, T. Ohgane, and T. Nishimura, “MIMO technologies for ultra high-speed wireless transmission,” IEICE B+ Plus, no. 11, pp. 32–38, 2009.
[10] A. Ghosh, R. Ratasuk, B. Mondal, N. Mangalvedhe, and T. Thomas, “LTE-advanced: next-generation wireless broadband technology,” IEEE Wireless Communications, vol. 17, no. 3, pp. 10–22, 2010.
[11] K. Zheng, L. Huang, G. Li, H. Cao, W. Wang, and M. Dohler, “Beyond 3G evolution,” IEEE Vehicular Technology Magazine, vol. 3, no. 2, pp. 30–36, 2008.
[12] C. J. Ahn, “Achievable throughput enhancement based on modified carrier interferometry for MIMO/OFDM,” Digital Signal Processing, vol. 20, no. 5, pp. 1447–1457, 2010.
[13] N. Gejoh and Y. Karasawa, “OFDM transmission characteristic in multipath environment where the delay spreading
exceeds the guard interval analyzed based on the ETP model,” *IEICE Transactions on Communications*, vol. J85-B, no. 11, pp. 1904–1912, 2002.

[14] S. Suyama, H. Suzuki, and K. Fukawa, “A MIMO-OFDM receiver employing the low-complexity turbo equalization in multipath environments with delay difference greater than the guard interval,” *IEICE Transactions on Communications*, vol. E88-B, no. 1, pp. 39–46, 2005.

[15] D. Van Nguyen, “Interference cancellation for MIMO-OFDM systems in the case of insufficient guard interval length,” in *1st International Conference on Communications and Electronics (ICCE '06)*, pp. 178–183, October 2006.

[16] Y. Ida, C. J. Ahn, T. Kamio, H. Fujisaka, and K. Haeiwa, “An interference cancellation scheme for TFI-OFDM in time-variant large delay spread channel,” *Radioengineering*, vol. 18, no. 1, pp. 75–82, 2009.
Research Article
A Power-Efficient Soft-Output Detector for Spatial-Multiplexing MIMO Communications

Hsiao-Chi Wang, Tung-Lin Liu, Yuan-Wei Wu, and Hsi-Pin Ma

Department of Electrical Engineering, National Tsing Hua University, Hsinchu 30013, Taiwan

Correspondence should be addressed to Hsi-Pin Ma, hp@ee.nthu.edu.tw

Received 15 June 2011; Revised 14 September 2011; Accepted 17 November 2011

Academic Editor: Zhiyuan Yan

Copyright © 2012 Hsiao-Chi Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

VLSI implementation of a configurable power-efficient MIMO detector is proposed to support 4 × 4 spatial multiplexing and modulation from QPSK to 64-QAM. A novel tree search algorithm is proposed to enable the detector to provide soft outputs and to be implemented in parallel and pipelined hardware architecture. The frame error rate (FER) of the detector approaches the quasi-optimal sphere decoder, with 0.5-dB degradation. Moreover, the proposed detector can operate at the optimal voltage under different configurations and detect/recover timing error at run time by a novel adaptive voltage scaling technique with double sampling circuitry. The proposed detector, using TSMC 0.18 μm single-poly six-metal CMOS process with a core area of 1.17 × 1.17 mm², provides fixed throughput of 45 Mbps in 64-QAM configuration, 120 Mbps in 16-QAM configuration, and 60 Mbps in QPSK configuration. The normalized power efficiency of the design for 64-QAM and 16-QAM configurations is 1.56 Mbps/mW and 2.53 Mbps/mW, respectively. Compared with the conservative margin-based design, the proposed design achieves a 48.8% power saving.

1. Introduction

Multiple-input multiple-output (MIMO) techniques in combination with high constellation orders have been identified as a promising approach to high spectral efficiency systems. Prominent detection in spatial multiplexing is essential for system performance. Maximum likelihood (ML) detection is the optimal method in spatial multiplexing systems. Sphere detection (SD) methods [1–3] compute the ML solution by taking into consideration only the lattice points inside the sphere with a given radius. Because a soft-in-soft-out error-correction-code (ECC) decoder [4] can have better error-correction performance than a usual hard ECC decoder, a soft-output MIMO detector, which cooperates with ECC better, is needed for coded communication system. The soft-output MIMO detector algorithms, such as list sphere decoder (LSD) [5] and soft single tree search sphere decoder (STS-SD) [6], have quasi-optimal performance with ECC. However, LSDs have huge complexity when list is extended, and STS-SD has variable throughput when channel condition is changed. The high computational intensity and the variable throughput characteristic of the iterative methods prevent current practical implementations from conforming the requirements for actual chip area, latency, and power consumption.

Fixed-complexity sphere decoder (FSD) algorithm [7] is another practical solution to MIMO detection. The FSD is similar to SD but with different search criteria. FSD performs a fixed tree search by visiting all nodes in the top level and just visiting one node in other levels to simplify the tree search in SD. Therefore, the complexity of FSD can be lower than SD. The FSD algorithm presents a quasi-ML performance and fixed complexity in an uncoded system. However, FSD is not compatible with powerful soft-input ECC decoders and needs some modifications. The work in [8] provides a good reference. In this paper, we reduced the visiting nodes by another way of node visiting distribution and found the local minima for soft-output under the tradeoff between performance and complexity using a modified FSD scheme. Furthermore, a novel method to simplify the SE enumeration [9] is proposed with the FSD. Therefore, the proposed FSD does not need to sort all points of the constellation.
To achieve more power saving, a full-pipelined parallel architecture is proposed according to the modification of the algorithm. The parallel design and adaptive voltage scaling technique can provide a power-efficient ASIC solution.

In this paper, a robust and power-efficient solution for spatial-multiplexing MIMO detection is proposed. The proposed soft MIMO detector can provide LLR output to ECC decoder and provide a turbo-MIMO solution. The proposed MIMO detector has the following features.

(a) Provision of High Order Modulations. The proposed MIMO detector supports multiple modulation configurations, including QPSK, 16-QAM, and 64-QAM modulation.

(b) Soft-Output Performance. With a modified tree search algorithm, the proposed MIMO detector, which provides soft-valued outputs, is compatible with soft-in-soft-out ECC decoders to attain enhanced detection performance.

(c) Parallel and Pipelined Architecture. Iterative sphere decoding methods prevent decoders from efficient hardware implementation. The proposed soft-output fixed-complexity sphere decoder (SFSD) retains the advantage of the fixed-complexity sphere decoder (FSD) [7] and therefore can be implemented in parallel and full pipelined designs to increase hardware efficiency.

(d) Fixed Throughput. General soft-output sphere decoding solutions only provide variable throughput, which makes imperfect use of hardware resources due to the sequential nature. The SFSD provides fixed throughput and achieves better performance than usual sphere decoders when the throughput is fixed. The maximum throughput is 120 Mbps in the proposed decoder.

(e) Error-Recovered Adaptive Voltage Scaling. A novel adaptive voltage scaling method is applied to the detector to reduce power dissipation. With double sampling circuitry, a timing error will be detected and recovered at run time. Therefore, an optimal voltage can be achieved and also keep the processing from functionality violation. With these techniques, a 48.8% saving in power is achieved.

(f) Configurable, Complexity-Efficient, and Power-Efficient Hardware Implementation. The configurable ASIC provides fixed throughput of 45 Mbps in 64-QAM configuration, 120 Mbps in 16-QAM configuration, and 60 Mbps in QPSK configuration. The normalized power efficiency is 1.56 Mbps/mW and 2.53 Mbps/mW for 64-QAM and 16-QAM configurations, respectively. The complexity efficient is 1.19 Mbps/K-gate for 16-QAM configuration.

The remainder of the paper is organized as follows. Section 2 reviews the conventional sphere decoding algorithm. The proposed SFSD algorithm with related simulation results is introduced in Section 3. Section 4 presents the logic design. Section 5 reports the hardware implementation. Finally the paper is concluded in Section 6.

2. Conventional Sphere Decoding Algorithm

Let us consider a MIMO system with \( M \) transmitting and \( N \) receiving antennas (\( N \geq M \)). \( M \) streams of \( Q \) bits of data, \( x_{j,b} \), \( j = 1, 2, \ldots, M \), and \( b = 1, 2, \ldots, Q \), are mapped to an \( M \times Q \)-dimensional transmitted symbol vector \( s = [s_1, s_2, \ldots, s_M]^T \), using \( 2^Q \)-QAM modulation. The received complex vector, \( y \), is given by

\[
y = Hs + n,
\]

where \( H \) is an \( N \times M \) channel matrix, which is assumed to be known in advance, and \( n \) is the complex Gaussian noise vector.

The a posteriori log-likelihood ratio (LLR) of the bit \( x_{j,b} \), conditioned on the received symbol vector, \( y \), provides a soft-in-soft-out detector information for decision and can be expressed as

\[
L\left( x_{j,b} \mid y \right) = \ln \frac{\Pr [x_{j,b} = +1 \mid y]}{\Pr [x_{j,b} = -1 \mid y]}.
\]

By Bayes’ theorem, the max-log approximation, and proof in [5, 6], (2) can be rewritten as

\[
L\left( x_{j,b} \mid y \right) \equiv \min_{s \in \delta_{j,b}^{(-1)}} ||y - Hs||^2 - \min_{s \in \delta_{j,b}^{(1)}} ||y - Hs||^2,
\]

where \( \delta_{j,b}^{(-1)} \) and \( \delta_{j,b}^{(1)} \) are search spaces, \( \{s \mid x_{j,b} = -1\} \) and \( \{s \mid x_{j,b} = 1\} \), respectively. The maximum-likelihood (ML) minima in (3) is expressed as

\[
s_{ML} = \arg \min_{s \in \mathcal{C}^M} ||y - Hs||^2,
\]

where \( \mathcal{C}^M \) is the set of constellation symbols in the \( M \)-dimensional complex space. Letting \( x_{j,b}^{ML} \) be the binary complement of the \( b \)th bit in the \( j \)th data symbol of \( s_{ML} \), the other minima in (3) can be expressed as

\[
s_{j,b}^{ML} = \arg \min_{s \in \mathcal{C}^M} ||y - Hs||^2.
\]

By means of the QR decomposition of the channel matrix \( H = QR \), (4) and (5) can be reformulated as

\[
s_{ML}^{(4)} = \arg \min_{s \in \mathcal{C}^M} ||\tilde{y} - Rs||^2,
\]

\[
s_{j,b}^{ML}^{(5)} = \arg \min_{s \in \mathcal{C}^M} ||\tilde{y} - Rs||^2,
\]

respectively, where \( \tilde{y} = Q^H y = Rs + \tilde{n} \), \( Q \) is an \( N \times M \) matrix with orthogonal unit norm columns, \( R \) is an \( M \times M \) upper-triangle matrix, and \( (\cdot)^H \) denotes the Hermitian matrix operator. Note that the noise term \( \tilde{n} = Q^H n \) keeps the same statistical properties as \( Q \) is an unitary matrix [1].
Table 1: Mean and variance of the distribution of the number of the visiting branches for each layer under SD ordering and FSD ordering at SNR = 10 dB in 4 × 4 MIMO 16-QAM System.

| Layer | SD ordering | FSD ordering |
|-------|-------------|--------------|
|       | Mean        | Variance     | Mean        | Variance     |
| 1     | 1.283783145 | 0.648927997 | 1.20332918  | 0.384272589  |
| 2     | 1.621236302 | 1.412882193 | 1.406256828 | 0.75118707   |
| 3     | 2.096801078 | 3.001800963 | 1.627991568 | 1.339530374  |
| 4     | 2.875270419 | 6.793185738 | 4.744884884 | 13.61550639  |

In addition, sorted QR decomposition (SQRD) [10] is applied, computing the Euclidean distance (ED) \( \| \tilde{y} - R s \| ^2 \) in (6) and (7) recursively as

\[
d_i(s') = d_{i+1}(s^{i+1}) + \left\| \tilde{y}_i - \sum_{j=1}^{M} R_{ij} s_j \right\|^2 ,
\]

where \( d_m(s^{m+1}) = 0, s^t = [s_1, s_{i+1}, \ldots, s_M]^T \), and \( \tilde{y}_i \) and \( R_{ij} \) are, respectively, elements in \( \tilde{y} \) and \( R \). Lattice points with partial Euclidean distance (PED), \( d_i(s') \), greater than the square of a given radius \( r \), are invalid. Therefore, the candidate search is confined to lattice points within a sphere to reduce the detection complexity. Nevertheless, the sequential search property and the variable computational complexity prevent the conventional SD from efficient hardware design.

### 3. Proposed Soft-Output FSD (SFSD)

#### 3.1. Algorithm Description

FSD algorithm [7] is another solution to MIMO detection, which is similar to SD but has two major differences.

(i) A fixed tree search is performed that FSD visits all nodes in the top level and just visits one node in other levels to simplify the tree search in SD.

(ii) The channel matrix ordering [11] is modified that the smallest column norm of channel matrix \( H \) is ordered in the first level of the tree.

The FSD algorithm presents a quasi-ML performance and fixed complexity in an uncoded system. However, FSD cannot be compatible with powerful soft-input ECC decoders.

For soft-output MIMO detection, the quasi-ML solution in (6) and the local minima in (7) are essential to calculate LLR. Therefore, the proposed FSD is a modification of the FSD by finding the local minima in (7) for soft-output. Accordingly, a soft-output SD requires more branches than the FSD. Monte Carlo simulations for the number of required visiting branches for SD and FSD algorithms were performed. Tables 1 and 2 show the mean and variance of the number of branches for a visited node at each layer in a 4 × 4 MIMO system using 16-QAM and 64-QAM, respectively. The distribution helps FSD algorithm to fix the number of branches in tree traversal. Both tables show higher mean and variance of visited branches in SD ordering compared with FSD ordering in all layers except layer 4. This can reduce the searching nodes and save computation complexity. By denoting the number of branches at the \( i \)th layer as \( n_i \), we can denote the distribution of branch number for 4 layers by \( n = [n_1, n_2, n_3, n_4] \). A diagram can show the idea in Figure 1. We can see that means in Tables 1 and 2 are in the descending order from the third level to the first level. Then we can find that the property of \( n_i \) is

\[
E[n_3] \geq E[n_2] \geq E[n_1].
\]

SFSD may follow the property and can be fixed to \( n = [2,3,4,P]^T \), \( n = [3,3,3,P]^T \), where \( P \) is equal to the number of the constellation points (2^Q) or other distributions which follow (9).

Figure 2 shows FER performance of different SFSD distributions with LLR maximum \( L_{\text{max}} = 16 \) in 4 × 4 MIMO 16-QAM system. The simulation distributions \( n = [n_1, n_2, n_3, n_4] \) on the figure are selected based on (9) and the mean/variance of the visiting branches in Tables 1 and 2. SD can be considered as the optimal performance here. We can see that the branches of node can affect the performance of SFSD, and more branches can have better performance directly. But too high distribution \( n \) can cause the tree search complexity to increase. Table 3 presents the total visited nodes in different SFSD distributions with 16-QAM modulation. The distribution \( [1,1,1,16]^T \) is used in the hard output FSD algorithm. The total visited nodes may represent the computation complexity of SFSD. We can find that the \( n = [1,2,2,16]^T \) SFSD needs the least branch expansion and has only 0.5 dB performance degeneration compared to SD. The \( n = [2,3,4,16]^T \) SFSD has better performance which degrades about 0.1 dB but has the highest complexity. It is a tradeoff between performance and complexity. Therefore, the branch distribution \( n = [1,2,2,2]^T \) for tree search has low complexity and is proposed to full-pipeline parallel hardware architecture implementation in the proposed SFSD. The top layer, \( n_4 \), is set to the number of all constellation points.

The usual sphere decoders adopt the SE enumeration [9], which sorts all of the constellation points by the Euclidean distances between the received signal \( y_i \) and itself. The point with the smallest distance has the first priority and enumerates others in ascending order of distance. Because proposed SFSD \( [1,2,2,P]^T \) only needs to enumerate the
two smallest points of constellation, the SE enumeration can be simplified here. Therefore, the proposed SFSD does not need to sort all points. The following is the presentation of simplified enumeration. Figure 3 is an example of the 16-QAM constellation. The red point is the point of received signal \( y_i \). The complex valued constellation can be separated into real part and imaginary part. The nearest point can be found by comparing with the value of the dotted lines in real part and imaginary part, respectively. Then the nearest point which is inside the green cycle is confirmed. The second nearest point of real part and imaginary part can be easily found by comparing with the values of the gray lines which include the nearest point. The two points \( 2a \) and \( 2b \) inside the blue cycles are the second nearest points of real part and imagine part, respectively. Then PED computations between those points and \( y_i \) are required. The distances between the nearest point and the received signal point in the real part and imaginary part are denoted as \( d_{R1} \) and \( d_{I1} \), respectively. The distances between the second nearest point and the received signal point in the real part and imaginary part are denoted as \( d_{R2} \) and \( d_{I2} \), respectively. Then, the second nearest point can be decided by \( \min\{d_{R1} + d_{I2}, d_{R2} + d_{I1}\} \). Only some comparisons help the SFSD to achieve the SE enumeration. Note that these PED computations of real part and imaginary part are equivalent to the PED computations of two points in the complex valued constellation. Therefore, the simplified SE enumeration just costs two PED computations.

The sequential design for tree search has variable throughput due to the unpredictable number of visited nodes, which causes hardware inefficiency. The early termination (ET) [6] can solve the problem. It confines the frame run time to \( N_{\text{frame}} \cdot D_{\text{avg}} \) and lets every frame have the same delay during \( N_{\text{frame}} \) MIMO detections. The function confines the maximum number of visited nodes to a value in each tree search as

\[
D_{\text{max}}(k) = N_{\text{frame}} \cdot D_{\text{avg}} - \sum_{i=1}^{k-1} D(i) - (N_{\text{frame}} - k)M,
\]

where \( D(i) \) denotes the number of visited nodes for the \( i \)-th MIMO symbol vector.

3.2. Simulation Results. The simulation environment is Rayleigh flat fading channel with AWGN. The algorithm is
evaluated in terms of the number of visited parent nodes in tree search and frame-error-rate (FER) performance. All simulations are performed in a 4 × 4 MIMO system with Gray mapping QAM constellation modulation. In the simulations, data are coded in a rate \( R = 1/2 \) convolutional code with constraint length 7 and a [133, 171] polynomial generator. A soft-input Viterbi decoder (max-log BCJR algorithm [12]) is employed in the receiver. A frame consists of \( N_{\text{frame}} = 64 \) MIMO symbols. In other words, a frame consists of randomly interleaved \( N_{\text{frame}} \cdot M \cdot Q \) bits after outer encoding.

Figure 4 illustrates the frame-error-rate (FER) performance of the optimal SD, STS-SD [6], and proposed SFSD. At 2%-FER, only a 0.5-dB loss is observed between the SFSD and the optimal SD. The performance degradation is even smaller between the SFSD and the STS-SD.

Recursive computation of the PED as (8) for tree search leads to considerable computational complexity and latency. Hence, the number of visited parent nodes in tree search is an indicator for searching complexity and latency. Figure 5 shows the comparison of the average number of visited parent nodes between the STS-SD algorithm [6] and the proposed SFSD. It can be seen that the proposed method can reduce 20% of the average number of visited parent nodes. Accordingly, the low complexity advantage of the SFSD over the STS-SD can be observed.

### 4. Logic Design

The proposed MIMO detector, shown in Figure 6, consists of a preprocessing and an SFSD block. The preprocessing block performs sorted QR decomposition (SQRD) and FSD ordering. The matrix \( R \) is decomposed by SQRD and consists of real valued diagonal elements, \( R_r \), and complex valued off-diagonal elements, \( R_c \). The computation of \( \tilde{y} = Q^H \cdot y \) is also performed in the block. FSD ordering chooses the column with the second smallest column norm to process in each iteration of the SQRD algorithm instead of the smallest column norm. The smallest column norm of channel matrix is ordered in the top level of tree; the other columns of channel are ordered to be decreased from right to left. In other words, the second top level has the greatest column norm, and the leaf level has the second smallest column norm. The SFSD block performs the proposed SFSD algorithm and adopts proper parallel design for high throughput and full pipeline design for high clock rate. Figure 7 shows the detailed block diagram of the proposed SFSD. The SFSD block consists of several PED modules (PED4, PED3, PED2, and PED1), a list administration unit (LAU), and an LLR module. The PED4, PED3, PED2, and PED1 modules calculate PEDs for layer 4, 3, 2, and 1, respectively. The LAU module compares and sorts the Euclidean distances (EDs) which are accumulated by the PED modules. After \( 2^Q \) times \( (2^Q \) is the constellation size) of sequential update checking, the LAU module outputs the calculated local minima \( \lambda \) for each layer and the ML solution \( x^\text{ML} \) (Figure 7). LLR computation module substrates the smallest distance value \( \lambda_k^\text{ML} \) from the local minimum distance values \( \lambda_k \) which corresponds to the local minima with binary complement of the \( k \)th bit of \( x^\text{ML} \) and computes each LLR.
value. The order of LLR values is not the same as the order of the original input data because SQRD preprocessing reorders the matrix $R$. Hence, the order recovery is also implemented in the LLR computation module.

4.1. Adaptive Voltage Scaling with Double Sampling. Dynamic voltage scaling (DVS) [13] is a common technique to adjust system voltage and reduce power consumption. Some methods have been proposed for pessimistic designs that require large safety margins [14]. Instead of using guard range from the safety margin, Razor [15, 16] proposed an in situ timing error detection and correction mechanism, which can overcome variation from the PVT and significantly reduce the power consumption. In Razor-based DVS, each flip-flop in the critical path is augmented with a shadow latch, which is triggered by a delayed clock. By comparing the data sampled by the main flip-flop and the shadow latch, a timing error can be detected and the corrupt value will be restored by the correct value in the shadow latch.

An adaptive voltage scaling with double sampling scheme is applied to the proposed SFSD. As shown in Figure 8, the main flop is clocked by clk and the delayed flop is clocked by clk_del, which is delayed with respect to clock clk. The delay between the clock edges of clk and clk_del is designed such that the correct value is obtained at the next clock edge of clk_del, even in the presence of unpredictability in the wire delay. Therefore, the delayed flop is designed to operate in an error-free manner. The XOR gate is used to compare the data captured by the main flop and delayed flop. When the outputs of the two flops differ, the signal errq is active to indicate a timing error and turn the main flop to the error mode. Afterward, the correct value captured by the delayed flop will be sent by the main flop in the next cycle, causing a one-cycle penalty for error recovery. Figure 9 shows the timing diagram of the error detection and recovery.

At each pipeline stage, an error control circuit presented in Figure 10 is added for generating suitable control signals when an error is detected. Let the number of bit-lines in the pipeline be $w$ lines. The XOR outputs (errq signals) generated at all the $w$ bit lines at each pipeline stage of the pipeline are ORed together and fed as an input to the error control circuit. For the Error-OR-tree, the error signals from all pipeline flip-flops are OR together to generate a unified error signal. If the design has many double sampling flip-flop, the fan-in of the OR gate can be very large, requiring OR signal to be pipelined. For error stabilization, in some cases, it is possible for error signal to become metastable. So we add two flip-flops at the global error output to overcome this problem.

4.2. Modified Cell-Based Flow. The timing paths of each submodule are analyzed after top-down synthesis. The critical paths of the target design are in PED1 and PED2 modules. Therefore, the double sampling flip-flops are inserted into the critical paths and ORed all the error signals on each double sampling pipeline stage. Figure 11 shows the block diagram of the SFSD after double sampling pipeline insertion.

5. Hardware Implementation

The proposed DVS soft output MIMO detector, using TSMC 0.18 μm single-poly six-metal (1P6M) CMOS technology, is
shown in Figure 12. Table 4 lists the ASIC specification. All the power and throughput information in Table 4 operate at 1.8 V and 120 MHz.

This design surpasses the previous researches [6, 17] for its high power efficiency. A comparison (with the proposed postlayout simulation results) is given in Table 5, where the power efficiency is normalized to mitigate the impact of different process factors and throughputs using

\[
\text{Normalized power efficiency} = \left(\frac{\text{Power} \times \left(\frac{1.8}{V_{DD}}\right)^2 \times 0.18 \times \frac{1}{\text{Throughput}}}{V_{DD}}\right)^{-1}.
\]

While normalizing power efficiency using (11), the nominal \(V_{DD}\) of the proposed design is 1.8 V, instead of the scaled voltage, to show the power-saving effect of the applied voltage scaling technique.

By scaling down the supply voltage, the power consumption of the design can be reduced, as shown in Figure 13.
Table 5: Comparison of MIMO detector implementations.

|                      | STS-SD [6] | MKSE [17] | SOSD [18] | This work |
|----------------------|------------|-----------|-----------|-----------|
| Fixed throughput     | No         | Yes       | No        | Yes       |
| Technology           | 0.25 μm    | 0.13 μm   | 0.35 μm   | 0.13 μm   | 0.18 μm |
| Max. frequency       | 71 MHz     | 100 MHz   | 200 MHz   | 198 MHz   | 120 MHz |
| (V_{DD} = 1.8 V)     |            |           |           |           |
| Gate count           | 56.8 K     | 97 K      | 91 K      | 350 K     | 100.6 K |
| Max. throughput      | 28.4 Mbps  | 106.6 Mbps| 53.3 Mbps | 287.9 Mbps| 120 Mbps|
| (16-QAM)             |            |           |           |           |
| Complexity efficiency| —          | —         | 0.59 Mbps/K-gate | 0.82 Mbps/K-gate | 1.19 Mbps/K-gate |
| Normalized power     | —          | —         | 0.20 Mbps/mW  | 1.88 Mbps/mW  | 2.53 Mbps/mW |
| efficiency (16-QAM)  | —          | —         |           |           |         |

The lowest supply voltage for error-free operations at 100-MHz clock rate for 64-QAM and 16-QAM modulation is 1.44 V and 1.26 V, leading to 34.7% and 49.22% power reduction, respectively. When voltage scales to 1.26 V for 64-QAM modulation or 1.08 V for 16-QAM modulation, timing error occurs on the critical path, which is pipelined by double sampling circuitry. The duplicate circuitry (delayed flip-flop) is triggered by the error signal and recovers the critical path from the timing error. While further scaling down the supply voltage, some subcritical paths may suffer from timing error and cannot be recovered. When the supply voltage decreases to 1.08 V for 64-QAM modulation, the functionality of the design fails due to incorrect data transferred by subcritical paths. With the proposed adaptive voltage scaling method with double sampling, the supply voltage can scale down to 1.08 V and 1.26 V, and the power reduction approaches 48.8% and 61.25%, for 16-QAM and 64-QAM modulation, respectively. Figure 14 shows the maximum clock rate and the current of the design in 16-QAM modulation. The power consumption ratio of each module is shown in Figure 15.
By using the error monitoring circuit, timing error information can be reported and conveyed to the voltage controller at run time to control the voltage regulator to power up/down the system. If the supply voltage is inadequate to support required timing of critical paths, the power controller will send a decision bit to indirectly control the voltage source which is provided by a DC-DC converter. The power management of the platform is completed through these mechanisms. The double sampling scheme shows well-behaved function, provides a solution to timing error detection and recovery, and accomplishes a robust DVS platform.

6. Conclusion

A power-efficient SFSD for MIMO detection is presented in the paper. The configurable SFSD with a novel tree search algorithm achieves soft-output decoding with fixed throughput of 120 Mbps in 16-QAM modulation. The FER of the detector approaches the optimal sphere decoder, with 0.5-dB degradation. A novel adaptive voltage scaling method with double sampling circuitry provides error detection/recovery and a 48.8% power saving.

Acknowledgments

The authors would like to thank the National Chip Implementation Center (CIC) of the National Applied Research Laboratories, Taiwan, for EDA tool support, as well as fabrication and measurement of the proposed chip. They would also like to thank the anonymous reviewers and editor for their valuable suggestions, which helped them improve this paper. This research is supported in part by the National Science Council, Taiwan, under Grants NSC-99-2221-E-007-050, NSC-99-2220-E-007-024, NSC-98-2219-E-007-003, supported in part by NTHU/ITRI joint research center, and supported in part by MediaTek-NTHU Joint Lab.

References

[1] M. O. Damen, H. E. Gamal, and G. Caire, “On maximum-likelihood detection and the search for the closest lattice point,” *IEEE Transactions on Information Theory*, vol. 49, no. 10, pp. 2389–2402, 2003.
[2] U. Fincke and M. Pohst, “Improved methods for calculating vectors of short length in a lattice, including a complexity analysis,” *Mathematics of Computation*, vol. 44, no. 5, pp. 463–471, 1985.
[3] A. Burg, M. Borgmann, M. Wenk, M. Zellweger, W. Fichtner, and H. Bölcskei, “VLSI Implementation of MIMO detection using the sphere decoding algorithm,” *IEEE Journal of Solid-State Circuits*, vol. 40, no. 7, pp. 1566–1577, 2005.
[4] J. Hagenauer and P. Hoeher, “Viterbi algorithm with soft-decision outputs and its applications,” in *Proceedings of the IEEE Global Telecommunications Conference & Exhibition (GLOBECOM ’89)*, pp. 1680–1686, November 1989.
[5] B. M. Hochwald and S. Ten Brink, “Achieving near-capacity on a multiple-antenna channel,” *IEEE Transactions on Communications*, vol. 51, no. 3, pp. 389–399, 2003.
[6] C. Studer, A. Burg, and H. Bölcskei, “Soft-output sphere decoding: algorithms and VLSI implementation,” *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 2, pp. 290–300, 2008.
[7] L. G. Barbero and J. S. Thompson, “Fixing the complexity of the sphere decoder for MIMO detection,” *IEEE Transactions on Wireless Communications*, vol. 7, no. 6, Article ID 4543065, pp. 2131–2142, 2008.
[8] L. G. Barbero, T. Ratnarajah, and C. Cowan, “A low-complexity soft-MIMO detector based on the fixed-complexity sphere decoder,” in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP ’08)*, pp. 2669–2672, April 2008.
[9] C. P. Schnorr and M. Euchner, “Lattice basis reduction: improved practical algorithms and solving subset sum problems,” *Mathematical Programming, Series B*, vol. 66, no. 2, pp. 181–199, 1994.
[10] D. Wübben, R. Böhnke, J. Rinas, V. Kühn, and K. D. Kammeyer, “Efficient algorithm for decoding layered spacetime codes,” *Electronics Letters*, vol. 37, no. 22, pp. 1348–1350, 2001.
[11] C. Hess, M. Wenk, A. Burg et al., “Reduced-complexity mimo detector with close-to-optimal error rate performance,” in *Proceedings of the 17th Great Lakes Symposium on VLSI (GLSVLSI ’07)*, pp. 200–203, March 2007.
[12] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, “Optimal decoding of linear codes for minimizing symbol error rate,” *IEEE Transactions on Information Theory*, vol. 20, no. 2, pp. 284–287, 1974.
[13] M. Horowitz, E. Alon, D. Patil, S. Naffziger, R. Kumar, and K. Bernstein, “Scaling, power, and the future of CMOS,” in *Proceedings of the IEEE International Electron Devices Meeting (IEDM ’05)*, pp. 9–15, December 2005.
[14] T. D. Burd, T. A. Pering, A. J. Stratakos, and R. W. Brodersen, “Dynamic voltage scaled microprocessor system,” *IEEE Journal of Solid-State Circuits*, vol. 35, no. 11, pp. 1571–1580, 2000.
[15] D. Ernst, N. S. Kim, S. Das et al., “Razor: a low-power pipeline based on circuit-level timing speculation,” in *Proceedings of the 36th Annual International Symposium on Microarchitecture*, pp. 7–18, 2003.
[16] S. Das, D. Roberts, S. Lee et al., “A self-tuning DVS processor using delay-error detection and correction,” *IEEE Journal of Solid-State Circuits*, vol. 41, no. 4, pp. 792–804, 2006.
[17] Z. Guo and P. Nilsson, “Algorithm and implementation of the K-best Sphere decoding for MIMO detection,” *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 3, pp. 491–503, 2006.
[18] C. H. Liao, T. P. Wang, and T. D. Chiueh, “A 74.8 mW soft-output detector IC for 8×8 spatial-multiplexing MIMO communications,” *IEEE Journal of Solid-State Circuits*, vol. 45, no. 2, Article ID 5405138, pp. 411–421, 2010.
Research Article

CP-Based SBHT-RLS Algorithms for Tracking Channel Estimates in Multicarrier Modulation Systems

H. Ali

School of Electrical Engineering and Computer Science, Faculty of Engineering and Built Environment, The University of Newcastle, Callaghan NSW2308, Australia

Correspondence should be addressed to H. Ali, hassan.ali@newcastle.edu.au

Received 18 July 2011; Accepted 3 October 2011

Academic Editor: Yin-Tsung Hwang

Copyright © 2012 H. Ali. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Cyclic prefix (CP) in multicarrier modulation systems has been considered as an alternative to the training sequences to track channel estimates. In this paper, two new algorithms are developed that exploit CP from their data detection part and employ systolic block Householder transformation recursive least squares (SBHT-RLS) algorithms for channel tracking in multicarrier systems. The new methods are compared with the existing CP exploiting correlation matrix based block RLS (CMB-RLS) channel tracking approach to outline their relative advantages. Aspects of computational complexity and parallel implementation are addressed, and the algorithms are tested in terms of their channel estimation and tracking capabilities. Performance of the algorithms is also evaluated for varying forgetting factor parameter values, constellation size, and word lengths. Floating-point and fixed-point simulations are tailored to illustrate pertinent tradeoffs.

1. Introduction

Over the last two decades, multicarrier modulation has received considerable interest for its use in wireless and wireline communication systems [1–4]. It has been adopted in many communication standards, including digital audio broadcasting (DAB) [5], digital video broadcasting (DVB) [6], high-speed modems over digital subscriber lines (xDSLs) [4], and local area mobile wireless broadband [7].

Most multicarrier systems use coherent detection of data symbols, which requires reliable estimation of channel at the receiver. Channel state information is also necessary for techniques such as channel shortening [8], adaptive modulation/loading, and/or power control [9]. In applications such as discrete multitone (DMT) xDSL [4], channel is estimated through some initial training process, and retraining is required to track the channel variation. To avoid the system overhead due to retraining and thus to track the channel more efficiently, in [10], a correlation matrix based block recursive least-squares (CMB-RLS) algorithm is proposed. The algorithm takes advantage of the inherent redundancy introduced by the cyclic prefix (CP) to blindly estimate the channel. In [11], performance of the algorithm is analyzed considering both the effect of channel noise and decision error. The algorithm is further explored in [12], where its performance is analyzed considering the impact of exponential forgetting factor values, constellation size, and channel nulls. Also, in [13], the method is used in single-carrier (SC) modulation with frequency domain equalization (FDE) to maintain both system performance and throughput.

While CP-based CMB-RLS approach is standard complaint, there are two basic problems that make it unsuitable for real-time implementation. First, it relies on computation of inverse of the correlation matrix Φ per time update. The computational cost of performing the required matrix inversion in real time can be prohibitively high for a system with a large channel length (To reduce the computational complexity and thus processing power, this inversion cannot be done recursively using Matrix inversion Lemma (such as in conventional RLS (CRLS) algorithm [14]).). Second, the direct inversion and recursive inversion approaches are known to severely limit parallelism and pipelining that can effectively be applied in the practical implementation.

Usually, to minimize the round off error, matrix inversions are done with general-purpose digital signal processing (DSP) devices/processors using floating-point arithmetic. A disadvantage of this approach, however, is severe processing
power limitation due to small number of floating-point processing units commonly available per device. Specialized hardware with high-processing power is therefore required to execute requisite computations in real time. An appealing alternative for implementation is not to do this inversion explicitly and solve the problem through a computationally cheaper approach that works directly with data matrix and is realizable on the systolic array architecture offering large amounts of parallelism for high-speed very large scale integration (VLSI) implementation. In VLSI implementation, floating-point arithmetic units are more complex than those of fixed-point arithmetic, involving extra hardware overhead and more clock cycles [15]. Hence, the bit-level systolic architecture must be implemented with fixed-point arithmetic.

The QR decomposition (QRD) approaches for RLS problem have played an important role in adaptive signal processing, adaptive equalization, and adaptive spectrum estimation [16]. It is generally agreed that QRD-RLS algorithms are one of the most promising RLS algorithms, due to their numerical stability [17, 18] and suitability for VLSI implementation [19, 20]. There are three approaches to QRD-RLS problem, namely, Givens rotation (GR), modified Gram-Schmidt (MGS), and Householder transformation (HT) method. These methods have been successfully applied to the development of the QRD-RLS systolic array [16, 21–24]. Because HT generally outperforms GR and MGS methods under finite precision computations (see the references in [16]), and in the context of our application the channel needs to be updated for each block input data matrix, we focus our attention to the QRD-RLS algorithm based on block HT. Notice that HT is a well-known rank-k update approach and is one of the most efficient methods to compute QRD (Rank-1 updating fast QRD-RLS algorithms (where QRD is updated after the original data matrix has been modified by the addition and deletion of a row or column) [14] are not suitable here in particular due to high throughput (here the term throughput is used to indicate total number of data vectors at the input of the RLS algorithm) and speed requirements.). In [24, 25], Liu et al. investigated one such QRD-RLS algorithm using block HT. The work in [24] describes the block HT implementation on a systolic array and its application to RLS algorithm called systolic block HT-RLS (SBHT-RLS). So far, SBHT-RLS is used in beamforming and linear prediction applications but has not been applied for channel tracking in high-throughput multicarrier applications. The algorithm is well known for its computational efficiency, very good numerical properties, and parallel processing implementation advantages.

In this paper, we develop two new CP exploiting SBHT-RLS approaches for adaptive channel estimation in multicarrier systems.

The first approach is based on SBHT-RLS approach of Liu et al. In its original form, the SBHT-RLS does not provide access to channel weights, as its use has been limited to the problem seeking an estimate of output error signal. In the context of our application, the proposed approach finds the channel explicitly. In order to differentiate between the two techniques, the new method will be referred to as CP-based Direct SBHT-RLS approach. The proposed scheme is computationally efficient and can be mapped to triangular systolic arrays for efficient parallel implementation. Unfortunately, the scheme suffers from a major drawback, namely, back substitution, which is a costly operation to perform in array structure [26, 27].

The second approach relies on inverse factorizations to calculate least squares channel coefficients (weight vector) without back substitution. This approach also employs SBHT to recursively update the channel coefficients and thus preserves the inherent stability property of SBHT-RLS approach. The derivation of the inverse factorization method in this paper is done by generalizing the Extended QRD-RLS algorithm to block RLS case [28]. For this reason, this method will be referred to as CP-based Extended SBHT-RLS approach. We underscore here that this simple and straightforward derivation is different than the previous challenging work on block RLS using inverse factorizations in [29, 30]. Computational complexity of this scheme is equivalent to the first proposed scheme, but unlike the first scheme it is fully amenable to VLSI implementation and also results in improved steady-state performance.

For the sake of brevity, in the rest of this paper, we refer to the CP based CMB-RLS as CPE1, Direct SBHT-RLS as CPE2, and Extended SBHT-RLS as CPE3. Also, for uniformity, we closely follow the notation that appears in [10].

The paper is organized as follows. In the next section, we provide an overview of the DMT system model [10]. Section 3 explains the newly proposed algorithms, followed by a discussion on their computational complexity and systolic array implementation in Section 4. In Section 5, illustrating floating- and fixed-point simulations are conducted, while conclusions are drawn in Section 6. Some results contained in this paper have been presented/accepted for presentation in [31, 32].

Notation. \((·)^T\), \((·)^*\), and \(E[·]\) denote transpose, complex conjugate, and expectation operation. The Matlab notation \(X(:, m : m')\) is used to to denote the submatrix of \(X\) that contains the columns \(m\) to \(m'\). \(x(m : n')\) denotes the subvector of \(x\) comprising of entries \(n\) through \(n'\). \(I\) denotes identity matrix of size \(n\), \(0\) denotes the all zeros matrix of appropriate dimensions, and \(j = \sqrt{-1}\). The meaning of other variables will be clear from the context.

2. System Model

We consider a high-speed DMT data transmission system over digital subscriber lines, shown in Figure 1. The system has \(m/2\) complex parallel subchannels and illustrates the typical CP based adaptive channel estimation task, which is our main concern in this paper. Let \(\{s_n\}\) represent the data sequence to be transmitted over the channel. This input data is buffered to blocks, and each data block is divided into \(m/2\) bit streams and then mapped to quadrature amplitude modulation (QAM) constellation points \(X_{i,k}, i = 0, \ldots, m/2−1\) at time \(k\). After \(m\)-point inverse fast Fourier transform (IFFT) on the \(k\)th DMT block \(X_k = [X_{0,k}, X_{1,k}, \ldots, X_{m−1,k}]^T\)
domain signal is \( x = \frac{1}{m} \) the first \( m \) samples. Where the prefix part \( y = k \) is appended in front of \( x \) before transmission through the channel \( H(z) = \sum_{l=0}^{z-l} h_{l,k}z^{-l} \), having impulse response \( h_k = [h_{0,k}, h_{1,k}, ..., h_{v,k}] \) of length \( v = v + 1 \). At the receiver, the prefix part \( y(f) = [y_{-v}, ..., y_{-1}]^T \) is removed.

The relationship between prefix part \( y_k(f) \) and the transmitted signal may be expressed as \([10]\)

\[
y_k(f) = A_k h_k + n_k(f), \tag{1}
\]

where

\[
A_k = \begin{bmatrix}
    x_{-v,0} & x_{m-1,0} & \cdots & x_{m-v,0} \\
    \vdots & \ddots & \ddots & \vdots \\
    x_{-1,0} & \cdots & x_{v,0} & x_{m-v,0} \\
\end{bmatrix} = [a_{0,k}, a_{1,k}, ..., a_{v,k}],
\]

(2)

\( a_{j,k} \) is the \( j \)th column of \( A_k \), \( n_k(f) = [n_{-v,0}, ..., n_{m,0}]^T \), and \( n_{i,k} \sim \mathcal{N}(0, \sigma^2) \) is the channel noise.

After the FFT operation on \( y_k = [y_{0,k}, y_{1,k}, ..., y_{m-1,k}]^T \), the demodulated signal is \( Y_k = [Y_{0,k}, Y_{1,k}, ..., Y_{m-1,k}]^T \). The CP removes interblock interference (IBI) between \( X_k \)'s. The received symbols can thus be written as

\[
Y_{i,k} = X_{i,k}H_{i,k} + N_{i,k}, \quad i = 0, ..., m - 1, \tag{3}
\]

where \( H_{i,k} = \sum_{l=0}^{\gamma} h_{l,k}e^{-j(2\pi l/m)} \) is the channel frequency response and \( N_{i,k} = (1/\sqrt{m}) \sum_{l=0}^{m-1} \eta_{l,k}e^{-j(2\pi l/m)} \sim \mathcal{N}(0, \sigma^2) \) is the noise of the \( i \)th channel.

To get the estimation of \( X_{i,k} \) from \( Y_{i,k} \), a one-tap minimum mean square error (MMSE) equalizer \( w_{i,k} = (\Gamma_i^{1/2} H_{i,k}^*)/(\Gamma_i ||H_{i,k}||^2 + \sigma^2) \), where \( i = 0, ..., m - 1 \) and \( \Gamma_i = E[||X_{i,k}||^2] \), is then employed at the \( i \)th channel. The estimated data is then \( \hat{X}_{i,k} = Y_{i,k}W_{i,k} \). The decision is then made on \( \hat{X}_{i,k} \) to get the final output \( \hat{X}_{i,k} = q(\hat{X}_{i,k}) \), where \( q(\cdot) \) is the decision operation.

3. CP-Based SBHT-RLS Algorithms

3.1. CP-Based Direct SBHT-RLS Algorithm (CPE2). Based on the CP data model \((1)\), we define \( n \times r \) weighted data matrix and the \( n \times 1 \) weighted received vector in a recursive manner as

\[
\hat{A}_k = \Lambda \begin{bmatrix}
    A_{k-(n-1)} \\
    \vdots \\
    A_k \\
\end{bmatrix} = \begin{bmatrix}
    \lambda^{1/2} \hat{A}_{k-1} \\
    \vdots \\
    \hat{A}_k \\
\end{bmatrix}, \tag{4}
\]

\[
\hat{Y}_k(f) = \Lambda \begin{bmatrix}
    \hat{Y}_{k-(n-1)} \\
    \vdots \\
    \hat{Y}_k \\
\end{bmatrix} = \begin{bmatrix}
    \lambda^{1/2} \hat{Y}_{k-1} \\
    \vdots \\
    \hat{Y}_k \\
\end{bmatrix}, \tag{5}
\]
where $\Lambda$ is an $nv \times nv$ block-diagonal forgetting matrix of the form

$$
\Lambda = \begin{bmatrix}
\lambda^{(n-1)/2} I_v & \cdots & 0 & 0 \\
\vdots & \ddots & \vdots & \vdots \\
0 & \cdots & \lambda^{1/2} I_v & 0 \\
0 & \cdots & 0 & I_v \\
\end{bmatrix}, 
$$

(6)

with forgetting factor across blocks $0 < \lambda \leq 1$. The forgetting factor $\lambda$ is incorporated in the scheme to avoid overflow in the processors as well as to facilitate nonstationary data updating [25].

Suppose that at the $(k - 1)$th update we have QRD

$$
Q_{k-1}A_{k-1} = \begin{bmatrix} R_{k-1} \\ 0 \end{bmatrix}, 
$$

(7)

where $Q_{k-1}$ is an $(n - 1)v \times (n - 1)v$ orthogonal matrix and $R_{k-1}$ is a $r \times r$ upper triangular matrix.

Now by denoting $Q_{k-1}A_{k-1} = \begin{bmatrix} Q_{k-1} \\ 0 \end{bmatrix}$, we then have

$$
\overline{Q}_{k-1}A_k = \begin{bmatrix} R_{k-1} \\ 0 \\ \overline{A}_k \end{bmatrix}, 
$$

(8)

A $n \times n$ HT matrix $T$ is of the form $T = I_n - \beta vv^T$, where $\beta = 2/\nu v^T v = 2/\|v\|^2$. When a vector $x = [x_1, x_2, \ldots, x_n]^T$ is multiplied by $T$, it is reflected in the hyperplane defined by span($v$)$^\perp$. Choosing $v = x \pm \|x\|e_1$, where $e_1 = [1, 0, 0, \ldots, 0]^T$, then $x$ is reflected onto $e_1$ by $T$ as: $Tx = \pm \|x\|e_1$.

A series of HTs are then used to zero out $A_k$ in the right-hand side of (8). Let $H_k = H_k^{(1)} H_k^{(2)} \cdots H_k^{(i)}$ (a sequence of r-ordered matrix multiplications), where $H_k^{(i)}$ denotes the $i$th HT matrix (which zeroes out $i$th column of updated $A_k$) given as

$$
H_k^{(i)} = \begin{bmatrix}
H_k^{(i),11} & 0 & H_k^{(i),12} \\
0 & I_{(n-1)v-r} & 0 \\
H_k^{(i),21} & 0 & H_k^{(i),22} \\
\end{bmatrix}, 
$$

(9)

where $H_k^{(i),11}$ is $r \times r$ identity matrix except for the $i$th diagonal entry, $H_k^{(i),12}$ is $r \times v$ zero matrix except for the $i$th row, $H_k^{(i),21}$ is $H_k^{(i),22}$ is symmetric $v \times v$ matrix.

It is thus we have $H_k \overline{Q}_{k-1}A_k = \begin{bmatrix} R_k \\ 0 \end{bmatrix}$ and $Q_k = H_k \overline{Q}_{k-1}$. Now with

$$
Q_k[\hat{A}_k | \hat{y}_k^{(f)}] = \begin{bmatrix} R_k & u_k \\ 0 & v_k \end{bmatrix}, 
$$

(10)

**Table 1: CP-based Direct SBHT-RLS algorithm (CPE2).**

**Input:** $y_k^{(f)}$, $y_{k-1}^{(f)}$, \ldots, $y_{k-2}^{(f)}(\nu)$ and $y_k$

**Known parameters:** $\Gamma_0$ and $\epsilon_i$

**Selecting parameters:** $\lambda$ (with $0 < \lambda \leq 1$)

**Initialization:** $k = 0$, an initial training process is used to initialize $\hat{h}_0$ and $R_0 = \delta I$ (with $0 < \delta \ll 1$ is small positive scalar).

**Algorithm:** $k = 1, 2, 3,$

(1) $\hat{h}_{k-1} = (1/\sqrt{\nu}) \sum_i R_{k-1}^i e^{j2\pi il/m}$, $i = 0, \ldots, m - 1$

(2) $w_{k-1} = (\Gamma_k^i/\|R_{k-1}^i\|^2 + \sigma_i^2)$, $i = 0, \ldots, m - 1$

(3) $X_{k-1} = Y_{k-1} w_{k-1}$, $i = 0, \ldots, m - 1$

(4) $\pi_k = (1/\sqrt{\nu}) \sum_{i=1}^{m} g(X_{k-1}^i) e^{j2\pi il/m}$, $i = m - v, \ldots, m - 1$

(5) $A_k = \sqrt{\lambda} R_{k-1} \hat{y}_k^{(f)} + \sqrt{\lambda} Y_{k-1}^{(f)}$ and $Y_k^{(f)}$

(6) $H_k \hat{A}_k | \hat{y}_k^{(f)} = \begin{bmatrix} R_k & u_k \\ 0 & v_k \end{bmatrix}$

with $H_k = H_k^{(1)} H_k^{(2)} \cdots H_k^{(i)}$, where $H_k^{(i)} = H_k^{(i),11} H_k^{(i),12} \cdots H_k^{(i),22}$, with $H_k^{(i)} = H_k^{(i),11} H_k^{(i),12} \cdots H_k^{(i),22}$, and $H_k^{(i),12}$ is $r \times v$ zero matrix except for the $i$th row, $H_k^{(i),21}$ is $H_k^{(i),22}$ is symmetric $v \times v$ matrix.

The optimal solution is thus obtained by solving the upper triangular system $R_k \hat{h}_k = u_k$ by back substitution operation as follows:

$$
\hat{h}_k = \frac{u_k - \sum_{i=1}^{v} r(i,i) \hat{h}_{k-i}}{r(i,i)}, \quad i = v, \ldots, 0. 
$$

(12)

The matrix $\hat{A}_{k-1}$ can be uniquely QR factorized only if it is full column rank (i.e., rank $A_{k-1} = r$). Therefore, the minimum number of rows in $A_{k-1}$ must be at least large as the number of columns. To satisfy this requirement and thus to reduce the number of received blocks needed by CPE2 (and CPE3 in Section 3.2), in step (10), we set

$$
\hat{A}_k = \begin{bmatrix} \lambda^{1/2} R_{k-1} \\ A_k \end{bmatrix},
$$

$$
\hat{y}_k^{(f)}(\nu) = y_{k-2}^{(f)} \hat{y}_k^{(f)}(\nu), \quad y_{k-1}^{(f)} \hat{y}_k^{(f)}(\nu) = \hat{y}_k^{(f)}(\nu),
$$

(13)

Based on the above discussion, CPE2 algorithm is summarized in Table 1.
and the new column to construct the formula

$$Q_k \begin{bmatrix} \sqrt{\lambda} y_{k-1}^T \\ y_k^T \\ \theta^T \end{bmatrix} = \begin{bmatrix} r_k \alpha_{k-1}^T \\ W_k^T \end{bmatrix}.$$  \hspace{1cm} (18)$$

Now by applying Lemma 1 to (18) yields

$$R_k^{-1} u_k + W_k v_k = R_{k-1}^{-1} u_{k-1}. \hspace{1cm} (19)$$

From (19), we establish a simple recursion to compute the channel vector

$$h_k = h_{k-1} - W_k v_k. \hspace{1cm} (20)$$

This recursion can be written in component form as

$$h_{ik} = h_{ik-1} - w_{ik} v_k, \quad i = 0, \ldots, v, \hspace{1cm} (21)$$

where $w_{ik}$ is the $i$th column of the matrix $W_k^T$.

Based on the above discussion, CPE3 is formulated in Table 2.

Remarks. (i) Both algorithms are initialized in a training mode, the algorithms then switch to a decision-directed mode for channel tracking. Note that, in step (1), based on the previous channel estimate $h_{k-1}$, the previous frequency response $\hat{H}_{k-1} = [H_{0,k-1}, \ldots, H_{m-1,k-1}]$ is computed. In step (2), $\hat{H}_{k-1}$ is then used to compute equalization coefficients. The decision-directed data vector $X_k$ is then computed in step (3). In step (4), symbol estimates are projected onto the finite alphabet (FA), and the estimated transmitted CP data $X_k^{(f)}$ is obtained by performing partial FFT on the decision-directed projected samples $X_k$. In steps (5) through (7), the new channel estimate is then obtained by treating the resulting symbol estimates as the known symbols. The process of alternating between channel and symbol estimation steps is applied repeatedly.

(ii) In [29], Sakai has derived a method for extracting weight coefficients based on the inverse factorization method of Pan and Plemmons [34] and Liu’s SBHT-RLS algorithm. The time updating formula for channel coefficients is obtained by first generalizing the inverse factorizations for the block case and then deriving a formula for updating the channel coefficients. The complicated and challenging derivation gets rid of matrix operations by exploiting the relation between a priori and posteriori error vectors. Based on [35] and suggested by its author, Sakai has also presented a simpler derivation for updating the channel vector in [30]. In contrast, in the above discussion, the same result is derived by following a straightforward approach by generalizing the Extended QRD-RLS algorithm of Yang and Bohme [28] to the block RLS case.

Table 2: CP-based Extended SBHT-RLS algorithm (CPE3).

| Input: $y_k^{(f)}, y_{k-1}^{(f)}, \ldots, y_{k-2}^{(f)}(v)$ and $Y_k$ |
| Known parameters: $\Gamma$ and $\sigma_i$ |
| Selecting parameters: $\lambda$ (with $0 < \lambda \leq 1$) |
| Initialization: $k = 0$, an initial training process is used to initialize $\hat{h}_0, R_0 = \delta I$. |
| $R_k^{T} = \delta^{-2} I$ (with $0 < \delta \ll 1$ is small positive scalar). |
| Algorithm: $k = 1, 2, 3, \ldots$ |
| (1) $\hat{H}_{k-1} = (1/\sqrt{m}) \sum_{i=0}^{m-1} \hat{H}_{i,k-1} e^{-j(2\pi m/i)}$, $i = 0, \ldots, m-1$ |
| (2) $w_{k-1} = (\Gamma/\lambda)^{-1} H_{i,k-1} / (\|H_{i,k-1}\|^2 + \sigma_i^2)$, $i = 0, \ldots, m-1$ |
| (3) $\hat{X}_k = Y_k w_{k-1}, i = 0, \ldots, m-1$ |
| (4) $X_k = (1/\sqrt{m}) \sum_{i=0}^{m-1} q(\hat{X}_{k,i}) e^{j(2\pi m/i)}$, $i = m - v, \ldots, 1$ |
| (5) $A_k = \begin{bmatrix} \sqrt{\lambda} R_{k-1}^{T} \hat{X}_k \\ \hat{A}_k \end{bmatrix} = \begin{bmatrix} R_k \alpha_{k-1}^T \\ 0 \end{bmatrix}$ |
| $y_k^{(f)} = \begin{bmatrix} \sqrt{\lambda} y_{k-1}^{(f)} \\ \hat{y}_k^{(f)} \end{bmatrix} = \begin{bmatrix} \sqrt{\lambda} y_{k-1}^{(f)} \\ y_k^{(f)} \end{bmatrix}$ |
| $H_k[ A_k, B_k, y_k^{(f)} ] = \begin{bmatrix} R_k \alpha_{k-1}^T \\ 0 \end{bmatrix}$ |
| $W_k^T v_k$, where $H_k = H_k^{(1)} H_k^{(2)} \ldots H_k^{(r)}$, with $H_k^{(p)} = I - e^{j(\beta-2\pi i/p)}$, $p = 2v+1$, $\beta = 2v\gamma$, $\gamma = \sqrt{x^2 + e_i^T e_i}$, $e_i = [1, 0, \ldots, 0]^T$, and $x = A_k(i : p, i)$ |
| (7) $h_k = h_{k-1} - W_k v_k$ |

We next define a lemma, known as the matrix factorization lemma [33] that is very elegant tool in the development of QRD-RLS algorithms.

Lemma 1. If $A$ and $B$ are any two $N \times M (N \leq M)$ matrices, then

$$A^T A = B^T B, \hspace{1cm} (15)$$

if and only if there exists an $N \times N$ unitary matrix $Q (Q^T Q = I)$ such that

$$QA = B. \hspace{1cm} (16)$$

Applying Lemma 1 to (14), we obtain

$$R_k^{T} R_k^{T} = R_{k-1}^{T} R_{k-1}^{T} = I_r. \hspace{1cm} (17)$$

This shows that $R_k^{T}$ obtained is the correct inverse transposition of $R_k^{T}$ and can be updated by using the same orthonormal transformation $Q_k$. Next, we combine the second column of (10) and the new column to construct the formula

$$Q_k \begin{bmatrix} \sqrt{\lambda} y_{k-1} \\ y_k \\ \theta \\ \phi \end{bmatrix} = \begin{bmatrix} R_k \alpha_{k-1}^T \\ 0 \end{bmatrix}.$$
4. Computational Complexity and Systolic Array Implementation

4.1. Computational Complexity. The CPE1, CPE2, and CPE3 algorithms are similar in the CP estimation part (i.e., steps (1) through (4)), we therefore compare their complexities in the channel estimation part. The CPE1 channel estimation stage requires $\Theta(r^2)$ computations to update $h$. In contrast, due to absence of any matrix inversion as opposed to CPE1, it is possible to implement channel estimation parts of both the algorithms with $\Theta(r^2)$ operations per time update. This indicates that the proposed algorithms are computationally superior than the CPE1.

4.2. Systolic Array Implementation. The detection part of both proposed algorithms (comprising of steps (1) through (4)) is particularly simple for which many efficient systolic array architectures have been proposed. We therefore limit our discussion to possible implementation architectures for channel estimation part of the proposed algorithms.

The systolic array implementation of channel estimation section of CPE2 and its processing cells are shown in Figure 2, where adaptive filtering triangular update part (comprising of step (6)) is realized on a triangular vectorial systolic array as in [24] for $R_k$ and $u_k$ extraction. It consists of two sections: the upper triangular array (shown in part (a) of Figure 2), which stores and updates $R_k$ and the right-hand column of cells (shown in part (b) of Figure 2), which stores and updates $u_k$. The input data are fed from top and propagate to the bottom of the array. The rotation angles are calculated in left boundary cells, and propagate from left to right. The resulting $R_k$ and $u_k$ updates in step (6) are subsequently used in the linear bidirectional systolic array section [19] (shown in part (c) of Figure 2) to obtain the channel estimate using back substitution operation. Unfortunately, a critical obstruction appears because the process of the triangular-updates runs from the upper-left corner to the lower-right corner of the array, while the process of the back substitution runs in exactly the opposite direction. It is therefore pipelining of the two steps (the triangular update and back substitution) that seems impossible on a triangular array. Back substitution may be implemented as a separate operation on a parallel two-dimensional array [36]. Nevertheless, the two-dimensional array can become quite large for long channel lengths, requiring a substantial area for VLSI implementation. On the other hand, comparatively simpler linear array structure shown in Figure 2 is highly sequential, thus involving more time delay due to increased clock cycles to compute the channel coefficients. For these reasons, the back substitution in CPE2 is a costly operation to perform.

The CPE3 approach involves a time recursive QR solution to compute the channel vector $h$. The channel estimation part of CPE3 algorithm can be implemented by a fully pipelined rhombic systolic array obtained by combining lower triangular array with an upper triangular array. This implementation has been performed by Sakai in [29, 30] and is reproduced in Figure 3. The components of $R_k$ are updated in the upper triangular part (a) of Figure 3. Also, the components of $u_k$ are updated in part (b) in the same fashion as the off-diagonal components of $R_k$, with the input data $y_k$ from the top of this column and the output $v_k$ from the bottom of this column. Notice that systolic implementation in upper section of part (c) is similar to that in part (a), except that the array is now lower triangular, and each element is divided with $\beta = \sqrt{\lambda}$ before updating, and the input to the array is provided from the top in the form of a zero vector. A systolic array performing (20) is shown in lower portion of part (c) of Figure 3, where the cells in the bottom line, shown by small circles; perform (20) for calculating the tap coefficients. Each column of the lower triangular array whose cells are shown by diamonds perform $R_k^{-1}$ updating. The cells also calculate each column of $U_k^T$, appearing from the last diamond cell. Notice that due to absence of back substitution, the CPE3 algorithm is rich in parallel operations and therefore leads to more efficient and simple implementation on systolic processors.

5. Simulation Results

In this section, floating-point and fixed-point simulation results are presented to examine and compare the performance of the CPE1, CPE2, and CPE3 approaches. All simulations were carried out in a typical asymmetric digital subscriber line (ADSL) environment with perfect block synchronization, FFT size $m = 512$, the CP length $\nu = 32$, $\lambda = 0.75$, $\delta = 1e^{-3}$, and 4-QAM constellation for modulation, unless otherwise stated. For a fair comparison, for CPE1 we set forgetting factor across blocks $\mu_1 = \lambda$ and forgetting factor within blocks $\mu_2 = 1$. The mismatch performance is evaluated by averaged mean-square-error (MSE) per subchannel $\text{err}_r = \sum_{i \in U} ||X_i - \hat{X}_i||/|U|$, where $U$ is the set of indexes corresponding to the $U$ used subchannles and $|U|$ is the number of all the used subchannels [10]. The transmit power of all used subchannels is same (i.e., $\sigma_0^2 = \sigma^2$) and the noise power was set such that $\text{SNR}= 30$ dB (a typical value of SNR in ADSL environments).

The discrete channel impulse response with transfer function $H_0(D)$ for carrier service loop area (CSA) loop # 1 was obtained from the Matlab DMTTEQ Toolbox [37] and sampled at 2.208 MHz. For simulation purposes, the shorter channel was generated by subsampling, $H_0(D)$ was perturbed to obtain another test channel $H_1(D)$ (to mimic small variation in $H_0(D)$). Corresponding frequency responses for the two test channels are shown in Figure 4. Initially, the channel transfer function is $H_0(D)$, which remains unchanged for the first 400 data blocks. At data block 401, the channel is switched from $H_0(D)$ to $H_1(D)$. For all adaptive schemes, only the first DMT symbol was sent and is mimicked $D$. The channel is switched from $H_0(D)$ to $H_1(D)$. For all adaptive schemes, only the first DMT symbol was sent and is mimicked $D$. The channel is switched from $H_0(D)$ to $H_1(D)$.

Example 1. Figure 5 shows typical learning curves of the three algorithms, with adaptation factor parameter $\lambda$ values
of 0.75 (top plots) and 0.55 (bottom plots), under double-precision floating-point implementation (using IEEE standard for floating-point arithmetic (IEEE 754)). It can be seen that all the schemes are able to converge and can track the channel variation. The learning curves of CPE1 and CPE2 are overlaid and both the algorithms converge faster than CPE3. As compared to CPE3, the two algorithms are also seen to have greater uneven performance. In contrast, although CPE3 convergence is slower, it is seen to demonstrate superior steady-state performance. A close examination of CPE2 algorithm shows that the back substitution operation involves decision-feedback computation of channel coefficients. If a channel coefficient suffers from an error, this error weights heavily in the estimation of the next and subsequent channel coefficients. The erroneous estimated channel causes the next detection error. This decision error further propagates and causes subsequent decision errors. Consequently, CPE2 encounters performance loss. In contrast, channel is recursively updated without back substitution in CPE3. CPE3 is therefore seen to yield better performance.

A close observation of top and bottom plots of Figure 5 also indicates that convergence rate and steady-state performance of the three algorithms can be improved by lowering the value of $\lambda$. The price paid in growth is uneven performance which can be reduced and thus numerical stability can be improved by increasing the data block size (i.e., with the increased CP length), while the system latency is increased.

**Example 2.** Without giving a rigorous stability analysis, we verify the stability of the CPE1, CPE2, and CPE3 algorithms experimentally through a long-time simulation with $5 \times 10^3$ data blocks (considerably large number of samples). Corresponding results in Figure 6 show that the three algorithms do not show any sign of divergence and have very stable performance.

**Example 3.** The more complex the modulation alphabet, the narrower the gap between the symbol decision space and the higher the probability of error in detecting the signal [38]. Since the three algorithms rely on the FA property of source symbols, high-performance degradation is expected as the constellation size increases. It is therefore the three algorithms that may not be suitable for rate adaptation. To verify this, in this simulation example, we repeat Example 1 with 16-QAM and 64-QAM constellation sizes. Corresponding simulation results in Figure 7 show that the three algorithms
Figure 3: Systolic array implementation of channel estimation section of CPE3 (using Householder transformations) with processing cell descriptions.

Figure 4: Frequency responses for channels $H_0(D)$ and $H_1(D)$. 
Figure 5: Effect of the forgetting factor $\lambda$ on performance analysis curves of CPE1, CPE2, and CPE3 algorithms when tracking channels $H_0(D)$ and $H_1(D)$: $\lambda = 0.75$ (top plots) and $\lambda = 0.55$ (bottom plots).

Figure 6: Stability performance comparison with $5 \times 10^3$ data blocks ($\lambda = 0.75$ and 4-QAM).

Figure 7: Effect of modulation constellation size on algorithm performance curves (16-QAM (top plots), 64-QAM (bottom plots), $\lambda = 0.75$).

Example 4. In this section, due to inherent parallelism and thus suitability for fixed-point VLSI implementation, we examine the fixed-point performance of the CPE2 and CPE3 algorithms with 16, 24, and 32 bit data word length $W_L$ implementations for both data and channel coefficients. These $W_L$s are selected as a reasonable approximation as these data lengths are suitable for many applications. For fixed-point simulations, routines in Matlab are developed to mimic the operations of fixed-point arithmetic, and all quantities in the algorithms are represented with finite bits. The fixed-point representation requires $W_L = (l_i \text{ bits for integer part}) + (l \text{ bits for the fractional part}) + (1 \text{ bit for sign})$. For real number $x$, its quantized value $x_q$ is obtained as follows. With $W_L$ bits, the largest integers that can be represented are $\pm 2^{(W_L-1)} - 1$. When the value of $x$ falls outside...
the interval \([+2^{(W_L-1)} - 1, -2^{(W_L-1)} - 1]\), the saturation occurs, and the \(x_i\) is then taken as one of the boundary values, \(+2^{(W_L-1)} - 1\) or \(-2^{(W_L-1)} - 1\). On the other hand, if \(x\) lies within the interval \([+2^{(W_L-1)} - 1, -2^{(W_L-1)} - 1]\), then the \(l_i\) bits are computed to represent the integer part of \(x\), and the remaining bits \(l\) are used to represent the fractional part of \(x\). It is important to note here that for the above choice of \(W_L\), the thresholds are sufficiently larger than signal values involved in both the algorithms. The quantizer is therefore always expected to operate on values that are much lesser than the boundary values, and therefore no saturation errors are expected. The only errors that are introduced by finite precision approximations are the round-off errors.

Figure 8 provides performance plots of CPE2 (top) and CPE3 (bottom) with different \(W_L\) choices and floating-point performance. From the performance curves, we infer that both algorithms are able to track the channel without numerical stability issues with \(W_L=24\) and 32. The performance curves with \(W_L\) of 16 bits indicate unacceptable performance or breakdown caused by quantization errors for both the algorithms. For both the algorithms, increasing \(W_L\) above 24 bits does not result in any improvement and performance curves of their 24-bit and floating-point implementations are overlaid (there is no visible difference). It is therefore 24-bit finite word implementation is a reasonable approximation of their floating-point computation.

6. Conclusion

In this paper, by using numerically robust block HTs, two CP-based adaptive channel estimation algorithms have been presented for multicarrier systems. Conceptually, the new schemes maintain the same spirit of the CP based CMB-RLS channel tracking scheme. More precisely, the basic idea is to utilize CP data from the data detection part for adaptive channel estimation. The new approaches achieve the same purpose by replacing the computationally expensive CMB-RLS channel estimation part with the computationally cheaper SBHT-RLS alternatives. Among the two schemes, the method called CP based Direct SBHT-RLS is based upon Liu’s algorithm in the channel estimation part but adaptively updates channel vector instead of the error vector. The second method called CP based Extended SBHT-RLS is based upon Sakai’s algorithm in the channel estimation part but uses an independent and simpler derivation.

Floating-point performance curves indicate that all the three schemes are able to converge and can track channel variation without any stability problems. CPE1 and CPE2 exhibit identical stable performance, whereas CPE3 outperforms both the CPE1 and CPE2 techniques. In contrast to CPE1, what is remarkable here is that the CPE2 and CPE3 algorithms achieve their performance at lower computational complexity, enhanced parallelism, and pipelining for systolic array/VLSI implementation. All the three algorithms are seen to converge faster and perform better with lower values of forgetting factor parameter \(\lambda\). Our simulation results suggest that such advantages come at the price of greater uneven performance. Hence, moderate values of forgetting factor would be preferred where a balance in both performance and stability is required. The three techniques also show reduction in performance with the increase in modulation constellation size. Hence, these techniques are more appealing when the constellation size is small and may not be suitable for rate adaptation. It is also shown that in terms of finite word length behavior, 24-bit finite word implementation is a reasonable approximation of their typical floating-point computation (In practice, the word lengths are optimized with respect to the actual system requirements (i.e., chip area, latency, power consumption, FFT size, throughput), noise, channel length, and desired acceptable performance.).

Systolic array structures that allow efficient parallel implementations of the schemes with VLSI technology in real time were considered. The CPE2 approach is partially concurrent due to costly back substitution operation, whereas, CPE3 approach is highly concurrent due to the absence of back substitution operation and therefore lead to more efficient implementation on systolic processors.

The methods proposed in this paper are well suited for applications where good numerical properties, computational saving, and parallel processing implementation...
advantages (with improved performance (in case of CPE3 only)) are desired. Although a real baseband DMT case is the main focus of this paper, the proposed approaches can also be applied to the complex baseband case (wireless multicarrier systems). In such case, a further improvement in performance is possible by including forward error correction (FEC) decoding in the reliable reconstruction of transmitted symbols. Future interesting directions include studying hardware implementation problems, fine grain implementation/architecture of processing elements to workout total cost of operators (adders, multipliers, dividers, memory elements (delay elements), etc.) and algorithm latencies, modifications of the schemes to achieve reduced complexity, performance improvement, and stable implementations with reduced word lengths.

Acknowledgment

The author wishes to express his sincere thanks to the reviewers for their constructive comments and useful suggestions towards improving this paper.

References

[1] Z. Wang and G. B. Giannakis, “Wireless multicarrier communications: where Fourier meets Shannon,” IEEE Signal Processing Magazine, vol. 17, no. 3, pp. 29–48, 2000.

[2] “IEEE Part II: wireless LAN medium access control (MAC) and physical layer (PHY) specifications: high speed physical layer in 5 GHz band,” IEEE Std. 802.11a-1999, September 1999.

[3] K. Van Acker, G. Leus, M. Moonen, O. van de Wiel, and T. Pollet, “Per tone equalization for DMT-based systems,” IEEE Transactions on Communications, vol. 49, no. 1, pp. 109–119, 2001.

[4] J. S. Chow, J. C. Tu, and J. M. Cioffi, “A discrete multitone transceiver system for HDSL applications,” IEEE Journal on Selected Areas in Communications, vol. 9, no. 6, pp. 895–908, 1991.

[5] “Radio broadcasting systems: digital audio broadcasting (DAB) to mobile, portable and fixed receivers,” ETSI ETS 300 401, 1.3.2 ed., 2000.

[6] “Digital video broadcasting (DVB): framing structure, channel coding and modulation for digital terrestrial television,” ETSI EN 300 744, 1.3.1 ed., 2000.

[7] R. van Nee, G. Awater, M. Morikura, H. Takanashi, M. Webster, and K. W. Halford, “New high-rate wireless LAN standards,” IEEE Communications Magazine, vol. 37, no. 12, pp. 82–88, 1999.

[8] P. J. W. Melsa, R. C. Younce, and C. E. Rohrs, “Impulse response shortening for discrete multitone transceivers,” IEEE Transactions on Communications, vol. 44, no. 12, pp. 1662–1672, 1996.

[9] P. S. Chow, J. M. Cioffi, and J. A. C. Bingham, “A Practical discrete multitone transceiver loading algorithm for data transmission over spectrally shaped channels,” IEEE Transactions on Communications, vol. 43, no. 234, pp. 773–775, 1995.

[10] X. Wang and K. J. R. Liu, “Adaptive channel estimation using cyclic prefix in multicarrier modulation system,” IEEE Communications Letters, vol. 3, no. 10, pp. 291–293, 1999.

[11] X. Wang and K. J. R. Liu, “Performance analysis for adaptive channel estimation exploiting cyclic prefix in multicarrier modulation systems,” IEEE Transactions on Communications, vol. 51, no. 1, pp. 94–105, 2003.

[12] H. Ali and E. P. Ling, “On the performance of CP based exponentially weighted block RLS channel estimation algorithm for OFDM systems,” in Proceedings of IEEE Pacific Rim Conference on Communications, Computers, and Signal Processing (PacRim ’09), pp. 135–139, Victoria, Canada, August 2009.

[13] W. A. Syafe, K. Nishijo, Y. Nagao, M. Kurokasi, and H. Ochi, “Adaptive channel estimation using cyclic prefix for single carrier wireless system with FDE,” in Proceedings of the 10th International Conference on Advanced Communication Technology, pp. 1032–1035, February 2008.

[14] D. G. Manolakis, V. K. Ingle, and S. M. Kogon, Statistical and Adaptive Signal Processing: Spectral Estimation, Signal Modeling, Adaptive Filtering and Array Processing, McGraw-Hill Education, Singapore, 2000.

[15] K. J. Raghunath and K. K. Parhi, “Finite-precision error analysis of QRD-RLS and STAR-RLS adaptive filters,” IEEE Transactions on Signal Processing, vol. 45, no. 5, pp. 1193–1209, 1997.

[16] C. F. T. Tang, K. J. R. Liu, S. F. Hsieh, and K. Yao, “VLSI algorithms and architectures for complex Householder transformation with applications to array processing,” The Journal of VLSI Signal Processing, vol. 4, no. 1, pp. 53–68, 1992.

[17] H. Leung and S. Haykin, “Stability of recursive QRD-LS algorithms using finite-precision systolic array implementation,” IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 37, no. 5, pp. 760–763, 1989.

[18] M. G. Siqueira and P. S. R. Diniz, “Infinite precision analysis of the QR-recursive least squares algorithm,” in Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS ’93), vol. 1, pp. 878–881, May 1993.

[19] S. Haykin, Adaptive Filter Theory, Prentice Hall, Upper Saddle River, NJ, USA, 1996.

[20] G. Lightbody, R. Woods, and R. Walke, “Design of a parameterizable silicon intellectual property core for QR-based RLS filtering,” IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 11, no. 4, pp. 639–678, 2003.

[21] J. G. McWhirter and T. J. Shepherd, “Systolic array processor for MVDR beamforming,” IEEE Proceedings. Part F, vol. 136, no. 2, pp. 75–80, 1989.

[22] C. R. Ward, P. J. Hargrave, and J. G. McWhirter, “A novel algorithm and architecture for adaptive digital beamforming,” IEEE Transactions on Antennas and Propagation, vol. 34, no. 3, pp. 338–346, 1986.

[23] S. Z. Kalson and K. Yao, “A class of least-squares filtering and identification algorithms with systolic array architectures,” IEEE Transactions on Information Theory, vol. 37, no. 1, pp. 43–52, 1991.

[24] K. J. R. Liu, S. F. Hsieh, and K. Yao, “Systolic block Householder transformation for RLS algorithm with two-level pipelined implementation,” IEEE Transactions on Signal Processing, vol. 40, no. 4, pp. 946–958, 1992.

[25] K. J. R. Liu, S. F. Hsieh, and K. Yao, “Recursive LS filtering using block Householder transformations,” in Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP ’90), vol. 3, pp. 1631–1634, Albuquerque, NM, USA, April 1990.
[26] A. Elnashar, S. Elnoubi, and H. A. El-Mikati, “Performance analysis of blind adaptive MOE multiuser receivers using inverse QRD-RLS algorithm,” IEEE Transactions on Circuits and Systems I, vol. 55, no. 1, pp. 398–411, 2008.

[27] S.-J. Chern and C.-Y. Chang, “Adaptive linearly constrained inverse QRD-RLS beamforming algorithm for moving jammers suppression,” IEEE Transactions on Antennas and Propagation, vol. 50, no. 8, pp. 1138–1150, 2002.

[28] B. Yang and J. F. Bohme, “Rotation-based RLS algorithms: unified derivations, numerical properties, and parallel implementations,” IEEE Transactions on Signal Processing, vol. 40, no. 5, pp. 1151–1167, 1992.

[29] H. Sakai, “A vectorized systolic array for block RLS using inverse factorizations,” in Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP ’92), vol. 4, pp. 233–236, March 1992.

[30] H. Sakai, “A Vectorized systolic array for parallel weight extraction of block RLS,” International Journal of Adaptive Control and Signal Processing, vol. 8, no. 5, pp. 475–482, 1994.

[31] H. Ali, “A cyclic prefix based adaptive channel estimation algorithm for multicarrier systems,” in Proceedings of the IEEE International Symposium on Signal Processing and Information Technology (ISSPIT ’10), Luxor, Egypt, December 2010.

[32] H. Ali, “A cyclic prefix based extended QRD-RLS algorithm using block Householder transformation for adaptive channel estimation in multicarrier systems,” in Proceedings of the 3rd International Conference on Signal Acquisition and Processing (ICSAp ’11), Singapore, February 2011.

[33] G. H. Golub and C. F. Van Loan, Matrix Computations, The Johns Hopkins University Press, 3rd edition, 1996.

[34] C. T. Pan and R. J. Plemmons, “Least squares modifications with inverse factorizations: parallel implications,” Journal of Computational and Applied Mathematics, vol. 27, no. 1-2, pp. 109–127, 1989.

[35] J. G. McWhirter, “Algorithmic engineering in adaptive signal processing,” IEE Proceedings. Part F, vol. 139, no. 3, pp. 226–232, 1992.

[36] S. Y. Kung, VLSI Array Processor, Prentice Hall, Englewood Cliffs, NJ, USA, 1988.

[37] G. Arslan, M. Ding, B. Lu, Z. Shen, and B. L. Evans, “DMTTEQ Toolbox,” The University of Texas at Austin, http://users.ece.utexas.edu/~bevans/projects/adsl/dmtteq/dmtteq.html.

[38] H. Ali, A. Doucet, and D. I. Amshah, “GSR: a new genetic algorithm for improving source and channel estimates,” IEEE Transactions on Circuits and Systems I, vol. 54, no. 5, pp. 1088–1098, 2007.
Research Article

Two-Dimensional Iterative Processing for DAB Receivers Based on Trellis-Decomposition

Wim J. van Houtum1,2 and Frans M. J. Willems2

1 Catena Radio Design, Science Park Eindhoven, Ekersrijt 5228, 5692 EG Son en Breugel, The Netherlands
2 Department of Electrical Engineering, Technische Universiteit Eindhoven, Den Dolech 2, Postbus 513, 5600 MB Eindhoven, The Netherlands

Correspondence should be addressed to Wim J. van Houtum, wvhoutum@catena.nl

Received 6 July 2011; Revised 14 October 2011; Accepted 17 October 2011

Academic Editor: Chirn Chye Boon

Copyright © 2012 W. J. van Houtum and F. M. J. Willems. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We investigate iterative trellis decoding techniques for DAB, with the objective of gaining from processing 2D-blocks in an OFDM scheme, that is, blocks based on the time and frequency dimension, and from trellis decomposition. Trellis-decomposition methods allow us to estimate the unknown channel phase since this phase relates to the sub-trellises. We will determine a-posteriori sub-trellis probabilities, and use these probabilities for weighting the a-posteriori symbol probabilities resulting from all the sub-trellises. Alternatively we can determine a dominant sub-trellis and use the a-posteriori symbol probabilities corresponding to this dominant sub-trellis. This dominant sub-trellis approach results in a significant complexity reduction. We will investigate both iterative and non-iterative methods. The advantage of non-iterative methods is that their forward-backward procedures are extremely simple; however, also their gain of 0.7 dB, relative to two-symbol differential detection (2SDD) at a BER of $10^{-4}$, is modest. Iterative procedures lead to the significantly larger gain of 3.7 dB at a BER of $10^{-4}$ for five iterations, where a part of this gain comes from 2D processing. Simulations of our iterative approach applied to the TU-6 (COST207) channel show that we get an improvement of 2.4 dB at a Doppler frequency of 10 Hz.

1. Introduction

1.1. Problem Description. Digital audio broadcasting (DAB) systems, DAB+ systems, and terrestrial-digital multimedia broadcasting (T-DMB) systems use orthogonal frequency division multiplexing (OFDM), for which every OFDM-subcarrier is modulated by $\pi/4$-Differentially Encoded-Quaternary PSK (DE-QPSK) [1].

Commonly used classical DAB receivers perform noncoherent 2SDD with soft-decision Viterbi decoding [2]. Noncoherent detection schemes like 2SDD are not optimal and can be improved by multisymbol differential detection (MSDD), which is a maximum likelihood procedure for finding a block of information symbols after observing a block of received symbols [3]. For very large numbers of observations, the performance of MSDD approaches the performance of ideal coherent detection of DE-QPSK, which is given, for example, [4–6]. Noncoherent MSDD can also be used if channel coding is applied in a noniterative way, see [7, 8].

If MSDD is combined with iterative (turbo) processing (parallel concatenated systems were first described by Berrou et al. [9], serial concatenation was developed by Benedetto and coinvestigators [10–12]), it needs to be improved to get a more acceptable complexity. We were motivated by a number of encouraging results on serial concatenation of convolutional encoding followed by differential decoding with turbo-like decoding techniques, also referred to as Turbo-DPSK. Turbo-DPSK was investigated for single-carrier transmission on AWGN channels in [13–18], as well as for time-varying channels in [19–23]. The main objective of these papers was to reduce the complexity of the inner decoder. Two main methods can be distinguished: first an explicit estimation of the channel phase followed by coherent detection, see
[19, 20], and for the 2D-case [24–26], or secondly by directly calculating the a posteriori probabilities of the information symbols as in [17, 18, 22], and for the 2D case [27–29].

We focus in the present paper on 2D processing, that is, in both the frequency and time domains. We will propose methods based on iteratively demodulating and decoding blocks of received symbols in a DAB-transmission stream. First we will; however, summarize other 2D approaches that are relevant to our work.

The work of ten Brink et al. in [24] on 2D phase-estimate methods can be regarded as an extension of the results of Hoheer and Lodge in [20] to the multicarrier case. Park et al. in [25] improved the hard-decision approach of ten Brink et al. by considering soft-decision. Both [24, 25] rely on pilot symbols, which are not present in DAB-transmission [1] unfortunately. Blind channel estimation techniques were proposed by Sanzi and Necker in [26]. They proposed a combination of the iterative scheme of ten Brink in [24] and a fast converging blind channel estimator based on higher-order asymmetrical modulation schemes, which are not used within a DAB-transmission [1].

To obtain a posteriori probabilities of the information symbols in a 2D setting, May, Rohling, and Haase in [27–29] considered iterative decoding schemes for multicarrier modulation with the soft-output Viterbi algorithm (SOVA, [30]). The SOVA was used for differential detection as well as for decoding of the convolutional code. They used in the coherent setting an estimate of the phase based on a block of three by three received symbols, which are adjacent in time and frequency direction. They proposed, for the coherent case, to use only the current received symbol to obtain a symbol metric for the SOVA innerdecoder, actually ignoring the differential encoding. For the incoherent case, they used a transition metric for the SOVA innerdecoder based on the current and previously received symbol. These a posteriori detection schemes produce approximations of the a posteriori probabilities. Procedures that focus on efficient computation of exact values can be found in [5] for the coherent case, but also in [18] for the incoherent case.

To reduce complexity we accept a small performance loss due to channel-phase discretization (see, e.g., Peleg et al. [17] and Chen et al. [22]) in this contribution, but apart from that we determine the exact a posteriori probabilities of the information symbols in a 2D setting. Our starting point will be the techniques proposed by Peleg et al. in [17]. We discretize the channel phase into a number of equispaced values, but do not allow the “side-step” transitions that were proposed by Peleg et al. to track small channel-phase variations. Then we calculate, in an efficient way, the a posteriori probabilities of the information symbols using the BCJR-algorithm [31] in a 2D setting, see also [32]. We will consider 2D blocks and trellis decomposition. Each 2D block consists of a number of adjacent subcarriers of a number of subsequent OFDM symbols. Focussing on 2D blocks was motivated by the fact that the channel coherence-time is typically limited to a small number of OFDM symbols, but also since DAB transmissions use time-multiplexing of services, which limits the number of OFDM symbols in a codeword. Extension in the subcarrier direction is required then to get reliable phase estimates. The trellis-decomposition method allow us to estimate the unknown channel-phase efficiently. This phase is related to subtrellises of which we can determine the a posteriori probabilities. With these probabilities we are able to chose a dominant subtrellis, which results in a significant complexity reduction.

Franceschini et al. [33] also use the idea of trellis-decomposition and subtrellises (multiple trellises), to focus on estimating channel parameters. Variation of these parameters is tackled by applying the so-called intermix intervals, in which special manipulations (mix-metric techniques) on the forward and backward metrics are performed. Since we cannot track channel variations here, we apply a 2D approach which is based on the assumption that there are independent channel realizations within distinct blocks. We will explain later, in Section 2.1, why we cannot track the channel phase.

1.2. Paper Outline. In this paper, we will focus both on iterative and noniterative decoding techniques for DAB-like systems. In the next section, we will give a short outline of the DAB system. In Section 3, we will start our analysis by considering noniterative methods for the single-carrier case and introduce trellis-decomposition with a dominant subtrellis approach. In Section 4, we expand our single-carrier methods to the multicarrier case and introduce a 2D-block approach for demodulation. Iterative methods based on serial concatenation of convolutional codes (SCCC) and trellis-decomposition with two dominant subtrellis approaches are considered in Section 5. In Section 6, we generalize our iterative methods for the single-carrier case to the multicarrier case with 2D-block demodulation. Results of applying our approach to a practical case are shown in Section 7. Finally, Section 8 draws conclusions on our decoding procedures based on 2D blocks and trellis-decomposition for DAB-like systems.

2. Description of a Digital Audio Broadcasting (DAB) System

2.1. Overview. Terrestrial digital broadcasting systems like DAB, DAB+, and T-DMB, all members of the “DAB family,” comprise a combination of convolutional coding (CC), interleaving, π/4-DE-QPSK modulation followed by OFDM, see Figure 1. Time multiplexing of the transmitted services allows the receiver to perform per service symbol processing [1], see Figure 2 where 1536 is the number of “active” OFDM subcarriers for a DAB-transmission in Mode-I [1]; hence the receiver can decode a certain service without having to process the OFDM symbols that do not correspond to this service. Consequently, only at particular time instants within a DAB transmission-frame a small number (usually up to four) of OFDM-symbols need to be processed. This results in “idle time” for the demodulation and decoding processes.

Note, that due to this “idle time,” the mix-metric techniques of [33] cannot be applied to DAB receivers. However, if all the transmitted services are decoded, and there is no idle time, mix-metric techniques could be a
that is to be transmitted via this subcarrier. The symbols \( b_n, n = 1, 2, \ldots, N \), assume values in the (offset) alphabet \( \mathcal{B} = \{ e^{j2\pi p/n^2}, p = 0, 1, 2, 3 \} \). The transmitted sequence \( \mathbf{s} = (s_0, s_1, \ldots, s_N) \) of length \( N + 1 \) follows from \( b \) by applying differential phase modulation, that is,

\[ s_n = b_n s_{n-1}, \quad n = 1, 2, \ldots, N, \tag{1} \]

where for the first symbol \( s_0 = 1 \).

2.4. OFDM in DAB. OFDM in DAB is realized using a \( B \)-point complex IFFT, where \( B \) is 256, 512, 1024, or 2048. To compute the \( n \)-th time-domain OFDM-symbol \( \mathbf{s}_m = (\tilde{s}_{1,n}, \tilde{s}_{2,n}, \ldots, \tilde{s}_{B,n}) \), we determine

\[ \tilde{s}_{t,n} = \frac{1}{\sqrt{B}} \sum_{m=1}^{B} s_{m,n} e^{j2\pi (t-1)(m-1)/B}, \quad \text{for } t = 1, 2, \ldots, B, \tag{2} \]

and \( s_{m,n} \) is the \( n \)-th differentially encoded symbol corresponding to the \( m \)-th subcarrier, or equivalently the \( m \)-th element in the \( n \)-th frequency-domain OFDM-symbol, see Figure 1. Note, that the IFFT is a computationally efficient inverse discrete Fourier transform (IDFT) for values of \( B \) that are powers of 2. To prevent Intersymbol interference (ISI) resulting from multipath reception, a cyclic prefix of length \( L_{cp} \) is added to the sequence \( \mathbf{s}_m \). This leads to the sequence \( \mathbf{s}_n = (\tilde{s}_{B-L_{cp}+1,n}, \ldots, \tilde{s}_{B,n}, \tilde{s}_{1,n}, \tilde{s}_{2,n}, \ldots, \tilde{s}_{B,n}) \) that is finally transmitted.

We assume that the channel is slowly varying with an impulse response shorter than the cyclic-prefix length. Moreover, we assume that the channel coherence bandwidth and coherence time span multiple OFDM subcarriers and multiple OFDM symbols. Therefore, the channel-phase and gain might be assumed to be fixed for a number of adjacent subcarriers and consecutive symbols. This is the assumption on which we base our investigations. The channel phase and gain are assumed constant (yet unknown to the receiver) over a 2D block of symbols, see Figure 3.

The receiver, in the case of perfect synchronization, removes the (received version of the) cyclic prefix, and then...
applies a $B$-point complex FFT on the time-domain received sequence $\tilde{r}_n = (\tilde{r}_{1,n}, \tilde{r}_{2,n}, \ldots, \tilde{r}_{B,n})$, which results in the $B$ received symbols

$$r_{m,n} = \frac{1}{\sqrt{B}} \sum_{t=1}^{B} \tilde{r}_{t,n} e^{-j2\pi(m-1)(t-1)/B}, \quad \text{for } m = 1, 2, \ldots, B.$$  (3)

OFDM reception can be regarded as parallel matched-filtering corresponding to $B$ complex orthogonal waveforms, one for each subcarrier. This results in a channel model, holding for a 2D block of symbols, that is, given by

$$r_{m,n} = |h| e^{j\phi} s_{m,n} + n_{m,n},$$  (4)

for some subsequent values of $n$ and $m$, where the channel gain $|h|$ and phase $\phi$ are unknown to the receiver. It should be noted that a phase rotation proportional to $m$, due to a time delay, is removed by linear phase correction (LPC). This technique modifies the phase of each OFDM subcarrier with an appropriate rotation based on the starting position (time delay) of the FFT window within the OFDM symbol. In practise, this delay can be determined quite accurately.

In the next subsection, we focus on a single subcarrier.

### 2.5. Incoherent Reception, Channel Gain Known to Receiver

The sequence $s$ that is transmitted via a certain subcarrier is now observed by the receiver as sequence $r = (r_0, r_1, \ldots, r_N)$. Note that compared to the previous subsection we have dropped the subscript $m$ here. Since it is relatively easy to estimate the channel gain, we assume here that it is perfectly known to the receiver, and to ease our analysis we take it to be one. The received sequence now relates to the transmitted sequence $s$ as follows:

$$r_n = e^{j\phi} s_n + n_n, \quad \text{for } n = 0, 1, \ldots, N,$$  (5)

where we assume that $n_n$ is circularly symmetric complex Gaussian with variance $\sigma^2$ per component. Basically we assume that the random channel phase $\phi$ is real-valued and uniform over $[0, 2\pi)$. This channel phase is fixed over all $N+1$ transmissions and unknown to the receiver.

Accepting a small performance loss as in, for example, Peleg et al. [17] and Chen et al. [22], we may assume that the channel-phase is discrete and uniform over 32 levels, which are uniformly spaced over $[0, 2\pi)$, hence

$$\Pr\left\{ \phi = \frac{l\pi}{16} \right\} = \frac{1}{32}, \quad \text{for } l = 0, 1, 2, \ldots, 31.$$  (6)

We will first study the situation in which we consider a uniformly chosen channel phase in a single subcarrier. Later we will also investigate the setting in which a uniformly chosen channel phase is moreover constant over a number of adjacent subcarriers.

### 2.6. Equivalence between DE-QPSK and $\pi/4$-DE-QPSK

It is well known and straightforward to show that the $\pi/4$-DE-QPSK modulation, which is performed in each of the subcarriers, is equivalent to DE-QPSK. To see this, we define for $n = 1, 2, \ldots, N$

$$a_n = b_n e^{-j\pi/4},$$  (7)

and for $n = 0, 1, \ldots, N$

$$x_n = s_n e^{-j\pi n/4},$$

$$y_n = r_n e^{-j\pi n/4},$$

$$w_n = n_n e^{-j\pi n/4}.$$  (8)

It now follows that $a_n \in A = \{e^{j\pi n/2}, p = 0, 1, 2, 3\}, x_0 = 1$, and

$$x_n = b_{n-1} s_n e^{-j\pi n/4},$$

$$y_n = e^{j\phi} s_n + n_n e^{-j\pi n/4} = e^{j\phi} x_n + w_n.$$  (9)

Now we may conclude that also $x_n \in A$ for all $n = 0, 1, \ldots, N$, and that $w_n$, just like $n_n$, is circularly symmetric complex Gaussian with variance $\sigma^2$ per component. Moreover, since $b$ is Gray-coded with respect to the interleaved code bits, so is $a = (a_1, a_2, \ldots, a_N)$. From now on, we will therefore focus on DE-QPSK.

### 3. Detection and Decoding: Single-Carrier Case, Noniterative

We will start by considering the single-carrier case. For some single subcarrier, we will discuss DE-QPSK modulation with incoherent reception. Based on trellis decoding techniques, we will determine the a posteriori symbol probabilities under the assumption that the (quantized) channel phase is uniform and unknown to the receiver. We also assume the transmitted symbols to be independent of each other and uniform.

#### 3.1. Trellis Representation, Subtrellises, Decomposition

In this section, we will focus on noniterative detection. We start our analysis by noting that if we define $z_n = x_n e^{j\phi}$ for $n = 0, 1, \ldots, N$, then, since $x_0 = 1$ and $\phi$ is uniform over $\{\pi l/16, l = 0, 1, \ldots, 31\}$, it follows that

$$\Pr\left\{ z_0 = e^{j\pi l/16} \right\} = \frac{1}{32}, \quad \text{for } l = 0, 1, \ldots, 31,$$  (10)

and $z_n \in \mathbb{Z} \triangleq \{e^{j\pi l/16}, l = 0, 1, \ldots, 31\}$. Moreover, for $n = 1, 2, \ldots, N$,

$$z_n = a_n z_{n-1},$$

where $\Pr\left\{ a_p = e^{j\pi p/2} \right\} = \frac{1}{4}, \quad \text{for } p = 0, 1, 2, 3.$  (11)

The variables $z_n$ for $n = 0, 1, \ldots, N$ can now be regarded as states in a trellis, and the independent uniformly distributed (iud) symbols $a_1, a_2, \ldots, a_N$ correspond to transitions between states. The resulting graphical representation of our trellis can be found in Figure 4.
If we would use the standard BCJR algorithm for computing the a posteriori symbol probabilities in the trellis in Figure 4, we have to do $32 \times 4$ multiplications in the forward pass, $32 \times 4$ multiplications in the backward pass, and $4 \times 32 \times 2$ multiplications and 4 normalizations in the combination pass, per trellis section, if the a priori probabilities are all equal. In total, this is 512 multiplications and 4 normalizations per trellis section. We suggest to focus only on multiplications and normalizations in this paper since additions have a smaller complexity than multiplications and normalizations. (In the log-domain, multiplications and normalizations are replaced by additions, and additions are typically approximated by maximizations. This would more or less suggest to consider multiplications, normalizations, as well as additions, but for reasons of simplicity we neglect the additions here.)

An important observation for our investigations is that the trellis can be seen to consist of eight subtrellises $T_0, T_1, \ldots, T_7$, that are not connected to each other. A similar observation was made by Chen et al. [22]. We will discuss connections between our work on the trellis decomposition and that of [22] later.

Subtrellis $T_s$ consists of states $z_n \in Z_s = \{e^{i\pi l/16}, l = s + 8p, p = 0, 1, 2, 3\}, \text{ for } s = 0, 1, \ldots, 7$. Figure 4 shows the entire subtrellis $T_0$, and the first section of subtrellis $T_1$ and of subtrellis $T_7$.

Note that for the likelihood $y_n(z_n)$ corresponding to some state $z_n \in Z$ for $n = 0, 1, \ldots, N$ in the trellis $T$ or in a subtrellis, we can write that

$$y_n(z_n) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{|y_n - z_n|^2}{2\sigma^2}\right). \quad (12)$$

3.2. Forward-Backward Algorithm, Subtrellises. In this subsection, we would like to focus on computing the a posteriori symbol probabilities $\Pr\{a_n \mid y_0, y_1, \ldots, y_N\}$ for all $n = 1, 2, \ldots, N$ and all values $a_n \in A$. It will be demonstrated that it is a relatively simple exercise to do this. We will show that the resulting a posteriori probability is a convex combination of the a posteriori probabilities corresponding to the eight subtrellises. Computing the a posteriori probabilities for each subtrellis is simple and can be done without performing the BCJR algorithm, as was demonstrated by Colavolpe [5]. The coefficients of the convex combination do not depend on the trellis section index $n$ and are quite easy to determine as we will show here.

3.2.1. Forward Recursion. In our forward pass, we focus on subtrellis $T_s$, for some $s \in \{0, 1, \ldots, 7\}$. For that subtrellis we find out how to compute all the $\alpha$’s in that subtrellis first. Starting from $\alpha_0(z_0) = 1/32$ for all $z_0 \in Z_s$, we can compute the $\alpha$’s recursively from

$$\alpha_n(z_n) = \sum_{s_n \in \{0, 1, \ldots, 7\}} \alpha_{n-1}(z_{n-1}) \frac{1}{4} y_{n-1}(z_{n-1}), \quad (13)$$

for $n = 1, 2, \ldots, N$ and $z_n \in Z_s$. The notation $(z, a) \rightarrow z'$ stands for all states $z$ and symbols $a$ that lead to next state $z'$.

**Lemma 1.** If for $n = 0, 1, \ldots, N$ we define $K_n(n) \triangleq \sum_{z_n \in Z} (1/4) y_n(z_n)$, then we have

$$\alpha_n(z_n) = \frac{1}{32} \prod_{i=0}^{n-1} K_i(i). \quad (14)$$

for $n = 0, 1, \ldots, N$ and $z_n \in Z_s$

**Proof.** Our proof is based on induction. Clearly for $n = 0$ the result holds. Now assume that $\alpha_{n-1}(z_{n-1}) = (1/32) \prod_{i=0}^{n-2} K_i(i)$ for $z_{n-1} \in Z_s$, then from (13) we obtain

$$\alpha_n(z_n) = \sum_{(z_{n-1}, a_n) = z_n} \left(\frac{1}{32} \prod_{i=0}^{n-2} K_i(i)\right) \frac{1}{4} y_{n-1}(z_{n-1})$$

$$= \frac{1}{32} \prod_{i=0}^{n-2} K_i(i) \sum_{z_{n-1} \in Z_s} \frac{1}{4} y_{n-1}(z_{n-1})$$

$$= \frac{1}{32} \prod_{i=0}^{n-1} K_i(i), \quad (15)$$

for all $z_n \in Z_s$. □

3.2.2. Backward Recursion. Also in the backward pass we first focus only on subtrellis $T_s$ for some $s$. In this subtrellis, we would like to compute the $\beta$’s. Taking $\beta_N(z_N) = y(z_N)$ for $z_N \in Z_s$, we can compute all other $\beta$’s from

$$\beta_n(z_n) = \sum_{a_{n+1}} \frac{1}{4} y_n(z_n) \beta_{n+1}(z_n a_{n+1}), \quad (16)$$

where again $n = 0, 1, \ldots, N - 1$ and $z_n \in Z_s$. 

---

**Figure 4:** Trellis representation of the states $z_0, z_1, \ldots, z_N$ and the differentially encoded symbols $a_1, a_2, \ldots, a_N$ in the incoherent case. An edge between two subsequent states indicates that a transition between these states is possible. Note that the trellis can be decomposed into eight unconnected subtrellises.
Lemma 2. Based on definition of $K_s(n) \triangleq \sum_{z_n \in Z_n} (1/4)y_n(z_n)$ for all $n = 0, 1, \ldots, N$, we get

$$\beta_n(z_n) = y_n(z_n) \prod_{i=n+1}^{N} K_s(i),$$

(17)

for $n = 0, 1, \ldots, N$ and all $z_n \in Z_s$.

Proof. Again our proof is based on induction. Note first that for $n = N$ the result holds. Now assume that $\beta_{n+1}(z_{n+1}) = y_{n+1}(z_{n+1}) \prod_{i=n+2}^{N} K_s(i)$, for $z_{n+1} \in Z_s$. Then

$$\beta_n(z_n) = \sum_{a_n \in A} \frac{1}{4} y_n(z_n) \left( y_{n+1}(z_n a_n) \prod_{i=n+2}^{N} K_s(i) \right)$$

$$= y_n(z_n) \sum_{z_n \in Z_n} \frac{1}{4} y_{n+1}(z_n a_n) \prod_{i=n+2}^{N} K_s(i)$$

$$= y_n(z_n) \prod_{i=n+1}^{N} K_s(i),$$

(18)

for all $z_n \in Z_s$.

3.3. Combination. To determine the a posteriori symbol probability for symbol value $a_n \in A$, we compute the joint probability and density

$$\Pr\{a_n | y \} = \sum_{z_n \in Z_n} a_n \beta_n(z_n)$$

$$= \sum_{s=0}^{N} \sum_{z_{n-1} \in Z_n} a_n \frac{1}{4} y_{n-1}(z_{n-1}) \beta_n(z_{n-1} a_n)$$

$$= \sum_{s=0}^{N} \sum_{z_{n-1} \in Z_n} a_n \frac{1}{4} y_{n-1}(z_{n-1}) \prod_{i=n+2}^{N} K_s(i)$$

$$= \sum_{s=0}^{N} \sum_{z_{n-1} \in Z_n} a_n \frac{1}{4} y_{n-1}(z_{n-1}) \prod_{i=n+2}^{N} K_s(i)$$

(19)

If we consider the “middle” term in (19), then we see that

$$\sum_{a_n} \left( \sum_{z_{n-1} \in Z_n} \frac{1}{4} y_{n-1}(z_{n-1}) \prod_{i=n+2}^{N} K_s(i) \right)$$

$$= \sum_{a_n} \left( \sum_{z_{n-1} \in Z_n} \frac{1}{4} y_{n-1}(z_{n-1}) \prod_{i=n+2}^{N} K_s(i) \right)$$

$$= 4K_s(n-1)K_s(n).$$

(20)

From this we may conclude that

$$p(y) = 7 \sum_{s=0}^{N} \Pr\{s\} p(y | s) = \sum_{s=0}^{N} \frac{1}{8} \prod_{i=0}^{N} K_s(i),$$

(21)

with

$$\Pr\{s\} = \frac{1}{8}, \quad \text{for} \ s = 0, 1, \ldots, 7,$$

$$p(y | s) = \prod_{i=0}^{N} K_s(i).$$

(22)

Now observing that

$$\Pr\{a_n | y\} = \frac{(1/8) \prod_{i=0}^{N} K_s(i)}{\sum_{s=0}^{N} (1/8) \prod_{i=0}^{N} K_s(i)},$$

(23)

$$\Pr\{a_n | y, s\} = \frac{\sum_{z_{n-1} \in Z_n} (1/4) y_{n-1}(z_{n-1}) \prod_{i=n+2}^{N} K_s(i)}{4K_s(n-1)K_s(n)}$$

(24)

for $s \in \{0, 1, \ldots, 7\}$ and $a_n \in A$, we can write that

$$\Pr\{a_n | y\} = \sum_{s=0}^{N} \Pr\{s | y\} \Pr\{a_n | y, s\}.$$  

(25)

The right-hand side of this equation can be interpreted as a convex combination of a posteriori symbol probabilities $\Pr\{a_n | y, s\}$, one for each subtrellis, where the weighting-coefficients are the a posteriori subtrellis probabilities $\Pr\{s | y\}$. An a posteriori subtrellis probability is the conditional probability that the discrete channel phase modulo 8 equals $s$ for some $s = 0, 1, \ldots, 7$ given $y$.

The demodulator that operates according to (25) has three tasks, first the eight weighting coefficients (23) have to be computed, then for each of the eight subtrellises for all symbol values $a_n \in A$ and all $n \in \{1, 2, \ldots, N\}$, the a posteriori symbol probabilities have to be computed. Finally the weighting (25) has to be done. Computing the weighting coefficient requires for each subtrellis $s \in \{0, 1, \ldots, 7\}$ the computation of the factors $K_s(n)$ for $n = 0, 1, \ldots, N$. These factors should then be multiplied and normalized to form $\Pr\{s | y\}$. For these computations, 8 multiplications per trellis section are needed. Computing the a posteriori symbol probabilities $\Pr\{a_n | y, s\}$ can be done efficiently by applying the Colavolpe [5] technique to each subtrellis. As in Colavolpe each such a posteriori symbol probability is based on only two received symbols $y_{n-1}$ and $y_n$ as is shown in (24). This avoids the use of the BCJR method in full generality and leads to significant complexity reductions, that is, only $8 \times 4 \times 4 = 128$ multiplications and $8 \times 4 = 32$ normalizations are needed per trellis section. The weighting operation requires $8 \times 4 = 32$ multiplications, and therefore in total this approach leads to $8 + 128 + 32 = 168$ multiplications and 32 normalizations, which is considerably less than what we need for full BCJR.

3.4. Dominant Subtrellis Approach. Equation (25) shows how the exact a posteriori symbol probabilities can be determined. If the a posteriori subtrellis probabilities are such that one of the probabilities dominates the other ones then weighting (25) can be approximated by

$$\Pr\{a_n | y\} \approx \Pr\{a_n | y, \hat{s}\}, \quad \text{with} \ \hat{s} = \arg \max_s \Pr\{s | y\}.$$  

(26)

Observe that this approach involves the computations of the a posteriori symbol probabilities, as described in (24), only for the dominant subtrellis $\hat{s}$. This requires $4 \times 4 = 16$ multiplications and 4 normalizations only per trellis section. Together with the computation of the weighting coefficients $8 + 16 = 24$ multiplications and 4 normalizations are necessary. Therefore, this reduces the number of multiplications with respect to full weighting by a factor of seven.
3.5. Simulations. We use in our simulations, just like Peleg et al. [17], the de facto industry standard $R_c = 1/2$ convolutional code with generator polynomials $g_0 = 133$ and $g_1 = 171$, which is equal to the convolutional code with puncturing index $PI = 8$ of Table 29 in [1, Section 11.1.2, page 131]. The DAB, DAB+, and T-DMB bit-reversal time interleaver and block frequency interleaver are modeled by a bitwise uniform block interleaver generated for each simulated code block of bits, hence, any permutation of the coded bits is a permissible interleaver and is selected with equal probability, as is done in [17].

The demodulator calculates, for each OFDM-subcarrier, the a posteriori probability given by (25) for $N + 1 = 2, 4, 8,$ and 32. The demodulator is followed by a convolutional decoder, which needs as input soft-decision information about the coded bits. Now, it follows from Gray mapping, that is, ideal LLRs, for a decomposed trellis for convolutional code with generator polynomials $g$, and 32 ideally

$$
\lambda_n^1 = \ln \left( \frac{e^{m(n)} + e^{m(3\pi/2)}}{e^{m(0)} + e^{m(\pi/2)}} \right), \quad \lambda_n^2 = \ln \left( \frac{e^{m(n/2)} + e^{m(\pi/2)}}{e^{m(0)} + e^{m(3\pi/2)}} \right),
$$

with symbol metric

$$
m(\phi) = \ln \left( \Pr \{ a_n = e^{i\phi} | y \} \right),
$$

and where $\lambda_n^1$ corresponds to bit $b_1$ and $\lambda_n^2$ to bit $b_2$.

Figure 5 shows the Bit-Error Rate (BER) performance with the so-called ideal LLRs for a decomposed trellis for trellis-length $N + 1 = 2, 4, 8,$ and 32. On the horizontal axis is the signal-to-noise ratio $E_b/N_0 = 1/(2\sigma^2)$. The demodulator operates according to (25).

We will compare the performance of this demodulator with that of two well-known procedures described in the literature: firstly, to "classical" DQPSK [35, Section 4.5-5, page 224], that is, two-symbol differential detection (2SDD). This leads to a posteriori symbol probabilities as in (9) in Divsalar and Simon [3], that is, to

$$
\Pr \{ a_n | y_n, y_{n-1} \} \propto I_0 \left( \frac{1}{\sigma^2} | y_n a_n + y_{n-1} | \right), \quad \text{for } a_n \in \mathcal{A},
$$

where $I_0(\cdot)$ is the zeroth-order modified Bessel function of the first kind. Secondly, we will compare our results to coherently detected DE-QPSK. We assume that the received sequence is perfectly derotated, that is, $\tilde{y} = ye^{-j\phi}$. Then the a posteriori symbol probabilities are given by

$$
\Pr \{ a_n | \tilde{y} \} \propto \sum_{x_{n-1} \in \mathcal{A}} \exp \left\{ \frac{1}{\sigma^2} \Re \left( x_{n-1} ( \tilde{y}_n a_n + \tilde{y}_{n-1} ) \right) \right\},
$$

for $a_n \in \mathcal{A}$, as described by Colavolpe [5]. Note that (31) is similar to (24) for $s = 0$.

The simulation results, which are shown in Figure 5, demonstrate that the BER performance curves of 2SDD and trellis length $N + 1 = 2$ are practically identical as we expect. Moreover, the coherent-detection curve and the curve for very large trellis sizes ($N \to \infty$) are very close. The small performance loss is due to discretizing the channel-phase with 32 levels. Furthermore, Figure 5 shows that (a) larger values of $N + 1$ result in performance closer to the coherent-detection performance, and, (b) for $N + 1 = 32$ ideally computed LLRs for a decomposed trellis perform quite close to coherent detection, that is, the difference in signal-to-noise ratio ($E_b/N_0$) is less than 0.15 dB at a BER of $10^{-4}$.

Next, in Figure 6, we turn to the dominant subtrellis approach, which is denoted by "Max" in the legend. We compare for trellis-length $N + 1 = 2, 8,$ and 32, the difference in performance between ideal LLRs based on the a posteriori probabilities given by (25) and the approximated LLRs based on the dominant-subtrellis a posteriori probabilities specified in (26). It can be seen from Figure 6 that (a) for larger $N + 1$, the difference between the exact and approximated LLRs becomes smaller, and (b) for $N + 1 = 32$, the difference between the ideal LLRs and the approximated LLRs is less than 0.1 dB.

3.6. Some Conclusions. Our simulations demonstrate that for trellis length $N + 1 = 32$, the ideal LLRs and the approximated LLRs have a performance quite close to that of coherent detection. The difference in signal-to-noise ratio is less than $0.25$ dB at a BER of $10^{-4}$ for the dominant-trellis approach. Therefore, if we focus on a BER of $10^{-4}$ for obtaining an acceptable performance, with single subcarrier transmission, we need a trellis length $N + 1 \geq 32$.

With a trellis length of $N + 1 \geq 32$ received symbols, the channel coherence time needs to be in the order of
There is a second reason for arguing that large values of detail in Section 7, where we study a typical urban channel. Not all service symbols process QPSK streams. Instead of using a single trellis with length does not match to processing more than four OFDM symbols. It would again be nice if we could decompose the question of jointly decoding a block (2D) of received symbols. We have seen that the trellis-length need to be as large as possible. For obtaining an acceptable performance, it must be larger than 32. This might not always be true. Therefore, we want to investigate if we would concentrate on what was received over several subcarriers.

Now we assume that in each subcarrier \( m = 1, 2, \ldots, M \), a sequence \( a_m = (a_{m,1}, a_{m,2}, \ldots, a_{m,N}) \) is conveyed using differential encoding. For the components of the transmitted sequence \( x_m = (x_{m,0}, x_{m,1}, \ldots, x_{m,N}) \), we can write

\[
x_{m,n} = a_{m,n} x_{m,n-1}
\]

and \( x_{m,0} = 1 \). We assume that the channel phase is constant over the block of symbols; therefore,

\[
y_{m,n} = e^{i\phi} x_{m,n} + w_{m,n,1}
\]

where \( \phi \in \{\pi/16, l = 0, 1, \ldots, 31\} \) and uniform just as before, and the noise variables \( w_{m,n} \) are circularly complex Gaussian with variance \( \sigma^2 \) per component. The output sequence corresponding to subcarrier \( m \) is denoted by \( y_m = (y_{m,0}, y_{m,1}, \ldots, y_{m,N}) \).

Just like in the single carrier case, we can determine the a posteriori subtrellis probabilities:

\[
\Pr[s | y_1, y_2, \ldots, y_M] = \frac{\Pr[s] p(y_1 | s) p(y_2 | s) \cdots p(y_M | s)}{\sum_{s'=0}^{N} \Pr[s'] p(y_1 | s') p(y_2 | s') \cdots p(y_M | s')},
\]

where \( \Pr[s] = 1/8 \) for \( s = 0, 1, \ldots, 7 \) and

\[
p(y_m | s) = \prod_{i=0}^{N} K_{m,s}(i),
\]

where \( K_{m,s}(n) \triangleq \sum_{z_n \in \mathbb{Z}} (1/4)^n y_{m,n}(z_n) \). Note that for the likelihood corresponding to some state \( z_n \) for \( n = 0, 1, \ldots, N \) in the trellis \( T \) or in a subtrellis, we can write that

\[
y_{m,n}(z_n) = \frac{1}{2\pi \sigma^2} \exp \left( -\frac{|y_{m,n} - z_n|^2}{2\sigma^2} \right).
\]

Now the a posteriori symbol probability for \( a_{m,n} \in \mathcal{A} \) can be written as

\[
\Pr[a_{m,n} | y_1, y_2, \ldots, y_M] = \sum_{s=0}^{7} \Pr[s | y_1, y_2, \ldots, y_M] \Pr[a_{m,n} | y_m, s],
\]

where

\[
\Pr[s | y_1, y_2, \ldots, y_M] = \frac{(1/8) \prod_{m=1}^{M} \prod_{i=0}^{N} K_{m,s}(i)}{\sum_{z_{n-1} \in \mathbb{Z}} (1/4)^n y_{m,n-1}(z_{n-1}) \prod_{i=0}^{N} K_{m,s}(i)},
\]

\[
\Pr[a_{m,n} | y_m, s] = \frac{\sum_{z_{n-1} \in \mathbb{Z}} (1/4)^n y_{m,n-1}(z_{n-1}) a_{m,n}}{4K_{m,s}(n-1)K_{m,s}(n)}
\]

for \( s \in \{0, 1, \ldots, 7\} \) and \( a_{m,n} \in \mathcal{A} \).

This suggests that the demodulator first determines the a posteriori subtrellis probabilities (weighting coefficients) using (38), for which the first \( 8 \times M \times (N + 1) \) \( K \)-factors have

\[\text{Figure 6: Bit-error performance for LLRs computed as in (25), that is, ideal LLRs, and approximated LLRs computed as in (26), for different trellis-lengths.}\]
to be computed. Using the weighting coefficients, the convex combination in (37) then leads to the a posteriori symbol probabilities. Finding the a posteriori symbol probabilities \( \Pr[\hat{a}_{m,n} \mid y_m,s] \) again can be done using the Colavolpe [5] method for each subcarrier and for each subtrellis, where again such an a posteriori symbol probability is based on only the two received symbols \( y_{m-1} \) and \( y_{m,n} \) as is shown in (39). Again the BCJR method in full generality is not needed, and the number of required multiplications and normalizations per trellis section is the same as in the single carrier case.

Equation (37) shows how the exact a posteriori symbol probabilities can be determined. Just like in the single-carrier case, if one of the posteriori subtrellis probabilities are such that one of the probabilities dominates the other, ones then weighting (37) can be approximated as follows:

\[
\Pr[a_{m,n} \mid y_1, y_2, \ldots, y_M] \approx \Pr[a_{m,n} \mid y_m, \hat{s}],
\]

(40)

with

\[
\hat{s} = \arg \max_s \Pr[s \mid y_1, y_2, \ldots, y_M].
\]

(41)

Again this approach involves the computations of the a posteriori symbol probabilities only for the dominant subtrellis \( \hat{s} \). The resulting number of multiplications and normalizations per trellis section is the same as for the single carrier case.

4.2. Simulations. In the previous section, we analyzed and simulated the single subcarrier approach. Here we will discuss the simulations corresponding to the multi-carrier method. We will again study the coded BER versus the signal-to-noise ratio \( E_b/N_0 = 1/(2\sigma^2) \). The BER performance for the ideal LLRs, based on a posteriori probabilities computed as in (37), is shown in Figure 7 with a fixed block size of \( M(N + 1) = 16 \). This fixed block-size is realized by the parameter pair values \( (M,N + 1) = (1,16), (2,8), (4,4), \) and \( (8,2) \). The detector operates according to (37). The performance of 2SDD and coherently detected DE-QPSK are shown as reference curves.

From Figure 7, it can be observed that a 2D decomposition with a shortest possible trellis-length of \( N + 1 = 2 \) and \( M = 8 \) adjacent subcarriers performs identical to the largest trellis-length \( N + 1 = 16 \) and \( M = 1 \) subcarrier that is, the single-carrier case. Intermediate cases also have an identical performance.

We do not show the results of the dominant subtrellis approach for the multi-carrier case here, since these results are identical to the corresponding results for the single-carrier case shown in Figure 6 in the previous section.

4.3. Conclusion Noniterative Decoding. Our investigations for the noniterative 2D-case show that we are very close to the performance of coherent detection of DE-QPSK even for small values of the trellis length \( N + 1 \), by processing simultaneously several subcarriers. A next question is whether we can do better than this. In the literature, see, for example, Peleg et al. in [17] and Chen et al. [22], it is demonstrated that iterative decoding techniques lead to good results for differential encoding. Therefore, in the sequel of this paper we study iterative decoding techniques for DAB-like streams, with a special focus again on 2D blocks.

5. Detection and Decoding: Single-Carrier Case, Iterative

In the following two sections, we consider iterative decoding procedures. Peleg and Shamai [13] first demonstrated that iterative techniques could increase the performance of the demodulation procedures of DE-QPSK streams significantly. We specialize their approach to DAB systems, and solve a problem connected to the, in practise quite small, length of the trellises for each subcarrier, by turning to 2D blocks for iterative demodulation.

5.1. Serial Concatenation. In the current section, we will investigate iterative decoding procedures for DAB-like systems, which are based on convolutional encoding, interleaving, and DE-QPSK modulation. If we consider DE-QPSK modulation as the inner coding method and convolutional encoding as the outer code, then it is obvious that we can apply techniques developed for serially concatenated coding systems here, see Figure 8. Serially concatenated turbo codes were proposed by Benedetto and Montorsi [10] and later investigated in more detail in Benedetto et al. [12]. Iterating between the DPSK-demodulator and convolutional decoder for the incoherent case was first suggested (for a single carrier) by Peleg and Shamai [13], Hoeher and Lodge [20] also applied iterative techniques to the incoherent case, but focussed on channel estimation, to be able to use coherent detection. For an overview of related results, all for the single-carrier case, see Chen et al. [22].
We will start in this section by considering the single carrier case and our aim is again to find out what we can gain from decomposing the trellis used in the demodulator into a part that corresponds to the channel phase and a part that relates to differential encoding. In the section that follows, we will consider the multi-carrier setting.

5.2. Peleg Approach. In this subsection, we investigate the forward backward procedures where we drop the assumption that the symbols \( s_n \), \( n = 1, 2, \ldots, N \), are uniformly distributed. Interleaving should still guarantee the independence of the symbols, however.

Just like Peleg et al. [17] we focus on the entire trellis \( T \). Note, however, that our trellis is different from that of [17], in which tracking of small channel phase variations is made possible by adding “side-step” transitions. We do not have such transitions in our trellis and, therefore, our trellis can be decomposed in eight unconnected subtrellises. In the next subsection, we take advantage of this decomposition; however, first we will consider the undecomposed trellis.

Again starting from \( a_0(z_0) = 1/32 \) for all \( z_0 \in Z \), we can compute the \( a \)'s recursively from

\[
a_n(z_n) = \sum_{z_{n-1}} \alpha_{n-1}(z_{n-1}) \Pr(a_n | z_{n-1}) y_{n-1}(z_{n-1}),
\]

for \( n = 1, 2, \ldots, N \) and \( z_n \in Z \). Also in the backward pass, we consider the entire trellis \( T \). Taking \( \beta_N(z_N) = y(z_N) \) for \( z_N \in Z \), we can compute all other \( \beta \)'s from

\[
\beta_n(z_n) = \sum_{z_{n+1} \in Z} \gamma_n(z_n) \Pr(a_{n+1} | z_{n+1}) \beta_{n+1}(z_{n+1}),
\]

where again \( n = 0, 1, \ldots, N - 1 \), and \( z_n \in Z \).

To determine the a posteriori symbol probability for symbol value \( a_n \in A \), we compute the joint probability and density

\[
\Pr(a_n | y, s) = \frac{\Pr(a_n | y) \Pr(s | y) \Pr(a_n | s)}{\Pr(s | y)}.
\]

This expression also tells us how the resulting extrinsic information can be determined. It can be checked, see Benedetto and Montorsi [10], that multiplying by the factors \( \Pr(a_n) \) in the a posteriori information (44) should be omitted for obtaining extrinsic information. The extrinsic information is now further processed by the convolutional decoder. The results of the iterative procedure are discussed in Section 5.5.

Using the standard BCJR algorithm for computing the extrinsic symbol probabilities in the trellis in Figure 4, since a priori symbol probabilities are non-uniform now, leads to \( 32 \times 4 \times 2 \) multiplications in the forward pass, \( 32 \times 4 \times 2 \) multiplications in the backward pass, and \( 32 \times 4 \times 2 \) multiplications and 4 normalizations in the combination pass for computing extrinsic information, per trellis section. In total, this is 768 multiplications and 4 normalizations per trellis section per iteration. In the next subsection, we investigate the decomposition of the demodulation trellis.

5.3. Trellis Decomposition. Here we investigate whether we can decompose the entire trellis for the case where the a priori probabilities are nonuniform. We are interested in decomposing (44) in such a way that we can write

\[
\Pr(a_n | y) = \sum_s \Pr(s, a_n | y) = \sum_s \Pr(s | y) \Pr(a_n | y, s),
\]

for all \( a_n \in A \). The question now is how to compute the a posteriori subtrellis probabilities \( \Pr(s | y) \) for \( s = 0, 1, \ldots, 7 \).

It can be shown that

\[
\Pr(s, y) = \frac{\sum_{z_0 \in Z} \beta_0(z_0)}{32},
\]

and therefore

\[
\Pr(s | y) = \frac{\sum_{z_0 \in Z} \beta_0(z_0)/32}{\sum_{r=0}^{7} \sum_{z_r \in Z} \beta_0(z_r)/32}.
\]

Now for each subtrellis, we can determine the a posteriori symbol probabilities using

\[
\Pr(a_n | y, s) \propto \Pr(a_n | s) p(y | a_n, s) = \Pr(a_n, s) \sum_{z_{n-1} \in Z} \alpha_{n-1}(z_{n-1}) y_{n-1}(z_{n-1}) \beta_n(z_{n-1} a_n),
\]

and by omitting the factor \( \Pr(a_n, s) \) in (48), the corresponding extrinsic information. Note that this approach requires a backward pass through the entire trellis \( T \), first to find the weighting probabilities \( \Pr(s | y) \), for \( s = 0, 1, \ldots, 7 \). This requires \( 32 \times (4 + 1) = 160 \) multiplications per trellis section observing that in (43), \( y_s(z_n) \) can be put in front of the summation sign. Then for all subtrellises \( T \), we do a forward pass (requiring \( 8 \times 4 \times 4 \times 2 = 256 \) multiplications per section) and then combine the results to obtain the extrinsic symbol probabilities \( \Pr(a_n | y, s) \) for that subtrellis (for which we need \( 8 \times 4 \times 4 \times 2 = 256 \) multiplications and \( 8 \times 4 = 32 \) normalizations per section). Finally these probabilities have to be weighted as in (45) which requires \( 8 \times 4 = 32 \) multiplications. In total, this results in 704 multiplications and 32 normalizations per iteration. It should be noted that decomposition of the trellis does not result in a significant complexity reduction with respect to the Peleg approach. In the next subsection, we will discuss an approach that gives a relevant complexity reduction, however.
5.4. Dominant Subtrellis Approaches. To achieve a complexity reduction, we investigate a method that is based on finding, at the start of a new iteration, the dominant subtrellis first and then do the forward-backward processing for demodulation only in this dominant subtrellis.

Finding the dominant subtrellis for an iteration is done based on the a posteriori subtrellis probabilities $Pr\{s \mid y\}$ that are computed before starting this iteration. Now assuming that one of the a posteriori subtrellis probabilities dominates the other ones, we can write

$$Pr\{a_n \mid y\} = Pr\{a_n \mid y, \hat{s}\}, \text{ with } \hat{s} = \arg\max_s Pr\{s \mid y\}. \tag{49}$$

This approach involves the computations of the a posteriori symbol probabilities (and corresponding extrinsic information) as described in (42), (43), and (44) only for the dominant subtrellis $\hat{s}$. Computing the a posteriori subtrellis probabilities for each iteration and then focussing only on the forward pass and combination computations is less complex than following the Peleg procedure. For the best subtrellis $T_5$, we do a forward pass ($4 \times 4 \times 2 = 32$ multiplications per trellis section) and then we combine the results to obtain the a posteriori (actually extrinsic) symbol probabilities $Pr\{a_n \mid y, \hat{s}\}$ for that subtrellis ($4 \times 4 \times 2 = 32$ multiplications and 4 normalizations per section). In total, we now need 224 multiplications and 4 normalizations per trellis section per iteration.

A second approach involves choosing the dominant subtrellis only once, before starting with the iterations. Since before starting the iterations the a priori probabilities $Pr\{a_n\} = 1/4$, that is, are all equal, the analysis in Section 3.4 applies. The a posteriori subtrellis probabilities can be computed as in (23). Now we do the iterations only in the subtrellis that was chosen initially. This approach requires 84 multiplications and 4 normalizations per trellis section per iteration and is therefore essentially less complex than the Peleg technique. In our simulations, we will only use this last technique when we address dominant subtrellises.

5.5. Simulations. We simulated the Peleg method described in [17] and determined the BER versus the signal-to-noise ratio $Eb/N0 = 1/(2\sigma^2)$. This BER performance is shown in Figure 9 for trellis lengths practically infinite, that is, $N \to \infty$ and ideal LLRs are based on the a posteriori probability given by (44). The BER performance is shown for $L = 1, 2, \ldots, 5$ iterations, where $L = 1$ stands for no iterations. Note that since we are using ideal LLRs and infinite trellis lengths, the corresponding curves shown in Figure 9 can be regarded as target curves for the iterative (single-carrier) case. In addition, also here, 2SD and coherently detected DE-QPSK curves are shown for reference. Not in the figure are the curves corresponding to the approach based on decomposing the trellis and using weighting as in (45). As expected, the performance of this approach shows no differences with the Peleg approach in (44). From Figure 9, it can be seen that for a BER $= 10^{-4}$ the improvement in required signal-to-noise ratio is $\approx 4.1$ dB after $L = 5$ iterations. Figure 9 also shows that improvement decreases with the number of iterations and that the first iteration yields the largest improvement. Similar results were obtained by Peleg et al. [17].

To see how the performance in the iterative case depends on the trellis length $N$, we simulated the Peleg approach for $N + 1 = 2, 4$, and 32, for $L = 5$ iterations. The results are in Figure 10. It can be seen that the “iterative coding gain” increases, as expected, with $N$ and that, for $N + 1 = 32$, the performance is already quite close to that of $N \to \infty$.

Finally, we compared for $N + 1 = 4$ and 32, the difference in BER between the exact LLRs based on the a posteriori (extrinsic) probability given by (44) or (45) and the approximated LLRs based on the a posteriori (extrinsic) probability given by (49). The results are shown
for each subcarrier \( m \) approximated LLRs becomes smaller and that for approach for di

![Figure 11: Bit-error performance with dominant subtrellis approach for different values \( N \) and \( L = 5 \) iterations.](image)

in Figure 11. We can conclude from Figure 11 that for larger \( N + 1 \), the difference in performance between the exact and approximated LLRs becomes smaller and that for \( N + 1 = 32 \) the difference between the ideal LLRs and the approximation versions, by selecting the dominant subtrellis before starting with the iteration process, is less than 0.3 dB.

6. Detection and Decoding: Multicarrier Case, Iterative

6.1. Trellis Decomposition. Just like in the noniterative multicarrier case, we do the processing based on trellis decomposition and focus on the computation of the a posteriori subtrellis probabilities:

\[
\Pr \{ s \mid y_1, y_2, \ldots, y_M \} = \frac{\Pr \{ s \mid y_1 \mid s \} \cdot \Pr \{ y_2 \mid s \} \cdot \ldots \cdot \Pr \{ y_M \mid s \}}{\sum_{s=0}^{N} \Pr \{ s \mid y_1 \mid s \} \cdot \Pr \{ y_2 \mid s \} \cdot \ldots \cdot \Pr \{ y_M \mid s \}}
\]

(50)

Note that \( \Pr \{ s \} = 1/8 \) for \( s = 0, 1, \ldots, 7 \) and therefore it follows from (46) that

\[
p(y_m \mid s) = \frac{\sum_{z_0 \in \mathbb{Z}} \beta_{m,0}(z_0)}{4},
\]

(51)

for each subcarrier \( m = 1, 2, \ldots, M \).

Now the a posteriori symbol probability for \( a_{m,n} \in \mathcal{A} \) can be written as in (37) that is,

\[
\Pr \{ a_{m,n} \mid y_1, y_2, \ldots, y_M \} = \sum_{s=0}^{N} \Pr \{ s \mid y_1, y_2, \ldots, y_M \} \Pr \{ a_{m,n} \mid y_m, s \},
\]

(52)

where \( \Pr \{ a_{m,n} \mid y_m, s \} \) is computed as given by (48) for \( s \in \{ 0, 1, \ldots, 7 \} \) and \( a_{m,n} \in \mathcal{A} \). From these a posteriori probabilities, we can compute the extrinsic information that is needed by the convolutional decoder. Computing extrinsic information is actually a little bit easier since it involves less multiplications.

This suggests that, for each iteration, the demodulator first determines the a posteriori subtrellis probabilities using (50), for which first a backward pass in each of the \( M \) trellises corresponding to the subcarriers is needed.

Using the weighting coefficients, the convex combination in (52) leads to the a posteriori symbol probabilities. Finding the a posteriori symbol probabilities \( \Pr \{ a_{m,n} \mid y_m, s \} \) should be done in the standard way, taking into account that the backward passes were already carried out.

6.2. Dominant Subtrellis Approach. Equation (52) shows how the exact a posteriori symbol probabilities can be determined, in each iteration. Just like in the single-carrier case, if the a posteriori symbol probabilities are such that one of the probabilities dominates the other ones, then convex combination (52) can be approximated as follows:

\[
\Pr \{ a_{m,n} \mid y_1, y_2, \ldots, y_M \} \approx \Pr \{ a_{m,n} \mid y_m, \hat{s} \},
\]

(53)

with

\[
\hat{s} = \arg \max_s \Pr \{ s \mid y_1, y_2, \ldots, y_M \}.
\]

(54)

Again this approach involves, in each iteration, the computations of the a posteriori symbol probabilities only for the dominant subtrellis \( \hat{s} \).

If we compute the dominant subtrellis only before the start of the iteration process, we obtain a significant complexity reduction since the analysis in Section 3.4 applies. Moreover, all iterations are done in the initially chosen subtrellis. The methods described here will be evaluated in the next subsection.

6.3. Simulations. We have seen before that in the noniterative multi-carrier case the performance was more or less determined by the size \( M(N + 1) \) of the block. If the channel cannot be assumed to be constant for large values of \( N + 1 \), we can always increase the number of subcarriers \( M \) if the frequency selectivity allows this. Note that keeping \( N + 1 \) small also has advantages related to service symbol processing [1]. Here the situation is slightly different as is demonstrated in Figure 12. Increasing \( M \) has a positive effect on the performance; however, since the trellis-length \( N + 1 \) remains constant (and is quite small), the effect of iterating is limited. We see, however, that by increasing \( M \) from 1 to 8 we get an improvement of roughly 0.7 dB.

Finally, we compare for \( N + 1 = 4 \) and 32, for \( M = 8 \), the difference between the performance of exact LLRs based on the a posteriori (extrinsic) probabilities given by (52) and the approximated LLRs based on the a posteriori (extrinsic) probabilities given by (53), see Figure 13. We can observe from Figure 13 that, as expected, the larger \( N + 1 \) is, the smaller the difference between the exact and approximated LLRs becomes. For \( N + 1 = 4 \), the difference between the ideal LLRs and the approximation, by selecting the dominant subtrellis before starting to iterate, is roughly 0.3 dB.
To investigate the performance in practise, we have used phase and fixed (unit) gain in our analysis and simulations. So far we have used AWGN channels with unknown channel.

**Figure 12:** Bit-error performance for the multi-carrier case for different values \( M \), where \( N + 1 = 4 \) and for \( L = 5 \) iterations.

We use our methods for DAB transmission in Mode-I, where the inverse subcarrier spacing \( T_u = 1 \) ms and where the cyclic-prefix period \( T_{cp} = 246 \mu s \).

Now, with these settings, the normalized Doppler rate \( f_d T_u \) is 0.01 and 0.02, respectively.

Note that to prevent ISI in an OFDM-scheme, the delay differences on separate propagation paths need to be less than the cyclic-prefix period \([2]\), that is, the channel impulse response length \( \tau_m \) must satisfy \( \tau_m \leq T_{cp} \). Within the DAB-system, \( T_{cp} \equiv (63/256) T_u < T_u/4 \) [1] and therefore the coherence-bandwidth \( B_c \approx 1/\tau_m > 4/(1/T_u) \), which is at least 4 OFDM-subcarriers. For Doppler frequency \( f_d = 20 \) Hz, the coherence-time \( T_c \approx 1/(2 f_d) = 25 \) ms, which is \( \approx 20 \) OFDM-symbols (including cyclic prefix).

The channel gain representative for a 2D block, where it is assumed to be constant, is estimated similar to \((8)\) in Chen et al. [22], that is,

\[
| \hat{h} |^2 = \max \left( \frac{1}{M(N+1)} \sum_{m=1, n=0}^{M, N} | y_{m,n} |^2 - 2\sigma^2, 0 \right). \quad (55)
\]

The results of our simulations with the TU-6 model are shown in Figure 14, where the solid lines show the results for \( f_d T_u = 0.01 \) and the dashed lines for \( f_d T_u = 0.02 \). We have results for \( N + 1 = 18 \) with \( M = 1 \) and for \( N + 1 = 4 \) with \( M = 8 \). We considered iterative procedures with \( L = 5 \) iterations. In our simulations, we used the dominant subtrellis approach, where we have chosen the dominant subtrellis before starting the iterations.

The value for \( N + 1 = 4 \) might be seen as a representative frame-size for services broadcasted by the DAB-family in transmission Mode I. In this mode, \( N + 1 = 18 \) is the maximum possible number of interleaved OFDM symbols. Note that \( N + 1 = 18 \) is close to the coherence-time of our TU-6 channel for a Doppler frequency of 20 Hz.

**Figure 14:** Bit error performance for the TU-6 COST-207 channel with the solid lines for \( f_d T_u = 0.01 \), the dashed lines for \( f_d T_u = 0.02 \), \( N + 1 = (4, 18) \), \( M = (8, 1) \), and \( L = 5 \) iterations.

---

**7. Performance for TU-6 Channel Model**

So far we have used AWGN channels with unknown channel phase and fixed (unit) gain in our analysis and simulations. To investigate the performance in practise, we have used the TU-6 (Typical Urban 6 taps) channel model defined in [36], which is commonly used to test DAB, DAB+, or T-DMB transmission. Two maximum Doppler frequencies are chosen, that is, \( f_d = 10 \) and 20 Hz, representing DAB transmission (in Band-III) movement speeds between transmitter and receiver of \( \approx 45 \) and \( \approx 90 \) km/h, respectively.
It can be concluded from Figure 14 that for \( N + 1 = 18 \) and \( M = 1 \), reliable transmission is not possible for the TU-6 channel with movement speeds of \( \approx 45 \) km/h and \( \approx 90 \) km/h. The 2D-decomposition approach, however, with \( N + 1 = 4 \) and \( M = 8 \), there is a considerable improvement of roughly 2.4 and 1.6 dB for 10 and 20 Hz, respectively, in required signal-to-noise ratio possible, compared to 2SDD.

8. Conclusions

We have investigated decoding procedures for DAB-like systems, focussing on trellis decoding and iterative techniques, with a special focus on obtaining an advantage from considering 2D blocks and trellis decomposition. These 2D blocks consist of the intersection of a number of subsequent OFDM symbols and a number of adjacent subcarriers. The idea to focus on blocks was motivated by the fact that the channel coherence time is typically limited to a small number of OFDM symbols, but also since per service symbol processing is used which limits the number of OFDM symbols in a codeword.

We have used trellis decomposition methods that allows us to estimate the unknown channel-phase modulo \( \pi/2 \). This channel phase relates to subtrellises of which we can determine the a posteriori probabilities. Using these probabilities we can weigh the contributions of all the subtrellises to compute the a posteriori symbol probabilities. We can also use these probabilities to chose a dominant subtrellis for providing us with these a posteriori symbol probabilities. Working with dominant subtrellises results in significant complexity reductions. A second important advantage of trellis-decomposition is that it allows us to process in an efficient way several subcarriers simultaneously.

We have first investigated noniterative methods. The advantage of these methods is that forward-backward procedures turned out to be extremely simple since we could use Colavolpe processing [5]. The drawback of these noniterative methods is, however, that their gain, relative to the standard 2SDD technique, is modest. Iterative procedures result in a significantly larger gain, however. In this context we must emphasize that part of this gain comes from the fact that we can do 2D processing.

Simulations for the noniterative AWGN case show that (a) trellis-lengths of \( N + 1 \geq 32 \) are required and (b) that 2D dominant subtrellis processing with \( M(N + 1) = 32 \) outperforms 2SDD by 0.7 dB at a BER of \( 10^{-4} \).

For the iterative AWGN case with \( L = 5 \) iterations, simulations show that 2D dominant subtrellis processing with \( M(N + 1) = 32 \), where \( N + 1 = 32 \) and \( M = 1 \), outperforms 2SDD by 3.7 dB at a BER of \( 10^{-4} \). However, simulations also reveal that with \( M(N + 1) = 32 \), where \( N + 1 = 4 \) and \( M = 8 \), the iterative coding gain is reduced to 2.5 dB, which is caused by the smaller value of \( N + 1 \).

On the other hand, iterative simulations for a practical setting (i.e., the TU-6 model) show that (a) with trellis-length \( N + 1 = 18 \) and \( M = 1 \) (one subcarrier) no reliable communication is possible, but that (b) with a modest trellis-length \( N + 1 = 4 \) and \( M = 8 \) subcarriers, the iterative coding advantage is maintained and that the gain is roughly 2.4 dB for 10 Hz Doppler frequency, and 1.6 dB for 20 Hz.

Acknowledgments

The authors would like to acknowledge Professor J. W. M. Bergmans, the management and the technical staff of Catena Radio Design B. V., and NXP-Research Eindhoven for their support to accomplish this work. Moreover, They thank the anonymous reviewers for their comments and valuable suggestions.

References

[1] European Standard (Telecommunications series), “Radio broadcasting systems; digital audio broadcasting (DAB) to mobile, portable and fixed receivers,” ETSI EN 300 401 V1.4.1 (2006-06), v1.4.1, June 2006.
[2] G. Zimmermann, M. Rosenberger, and S. Dostert, “Theoretical bit error rate for uncoded and coded data transmission in digital audio broadcasting,” in Proceedings of the IEEE International Conference on Converging Technologies for Tomorrow’s Applications (ICC ’96), vol. 1, pp. 297–301, June 1996.
[3] D. Divsalar and M. K. Simon, “Multiple-symbol differential detection of MPSK,” IEEE Transactions on Communications, 38, no. 3, pp. 300–308, 1990.
[4] G. E. Bottomley, H. Arslan, R. Ramesh, and G. Brisまま, “Coherent MAP detection of DQPSK bits,” IEEE Communications Letters, 4, no. 11, pp. 354–356, 2000.
[5] G. Colavolpe, “Classical coherent receivers for differentially encoded M-PSK are optimal,” IEEE Communications Letters, 8, no. 4, pp. 211–213, 2004.
[6] W. J. Van Houtum and F. M. J. Willems, “A-posteriori symbol probabilities and log-likelihood ratios for coherently detected \( \pi/4-DE-QPSK,” IEEE Communications Letters, 15, no. 2, pp. 160–162, 2011.
[7] D. Divsalar and M. K. Simon, and M. Shahshahani, “The performance of trellis-coded MDPSK with multiple symbol detection,” IEEE Transactions on Communications, 38, no. 9, pp. 1391–1403, 1990.
[8] 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 Communications Letters, 42, no. 1, pp. 76–89, 1994.
[9] C. Berrou, A. Glavieux, and P. Thitimajshima, “Near shannon limit error correcting coding and decoding: turbo-codes. 1,” in Proceedings of the IEEE International Conference on Communications (ICC ‘93), vol. 2, pp. 1064–1070, Geneva, Switzerland, May 1993.
[10] S. Benedetto and G. Montorsi, “Iterative decoding of serially concatenated convolutional codes,” Electronics Letters, vol. 32, no. 13, pp. 1186–1188, 1996.
[11] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, “Serial concatenation of interleaved codes: performance analysis, design, and iterative decoding,” Tech. Rep. 42-126, TDA Progress, August 1996, pp.1–26.
[12] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, “Serial concatenation of interleaved codes: performance analysis, design, and iterative decoding,” IEEE Transactions on Information Theory, vol. 44, no. 3, pp. 909–926, 1998.
[13] M. Peleg and S. Shamai, “Iterative decoding of coded and interleaved noncoherent multiple symbol detected DPSK,” Electronics Letters, vol. 33, no. 12, pp. 1018–1020, 1997.

[14] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, “A soft-input soft-output APP module for iterative decoding of concatenated codes,” IEEE Communications Letters, vol. 1, no. 1, pp. 22–24, 1997.

[15] S. ten Brink, J. Speidel, and R. H. Yan, “Iterative demapping for QPSK modulation,” Electronics Letters, vol. 34, no. 15, pp. 1459–1460, 1998.

[16] I. D. Marsland and P. T. Mathiopoulos, “On the performance of iterative noncoherent detection of coded M-PSK signals,” IEEE Transactions on Communications, vol. 48, no. 4, pp. 588–596, 2000.

[17] M. Peleg, S. Shamai, and S. Galán, “Iterative decoding for coded noncoherent MPSK communications over phase-noisy AWGN channel,” IEE Proceedings: Communications, vol. 147, no. 2, pp. 87–95, 2000.

[18] G. Colavolpe, G. Ferrari, and R. Raheli, “Noncoherent iterative (turbo) decoding,” IEEE Transactions on Communications, vol. 48, no. 9, pp. 1488–1498, 2000.

[19] P. Hoeher and J. Lodge, “Iterative differential PSK demodulation and channel decoding,” in Proceedings of the IEEE International Symposium on Information Theory, p. 425, August 1998.

[20] P. Hoeher and J. Lodge, “‘Turbo DPSK’: iterative differential PSK demodulation and channel decoding,” IEEE Transactions on Communications, vol. 47, no. 6, pp. 837–843, 1999.

[21] L. H. J. Lampe and R. Schober, “Iterative decision-feedback differential demodulation of bit-interleaved coded MDPSSK for flat Rayleigh fading channels,” IEEE Transactions on Communications, vol. 49, no. 7, pp. 1176–1184, 2001.

[22] R.-R. Chen, R. Koetter, U. Madhow, and D. Agrawal, “Joint noncoherent demodulation and decoding for the block fading channel: a practical framework for approaching Shannon capacity,” IEEE Transactions on Communications, vol. 51, no. 10, pp. 1676–1689, 2003.

[23] V. Pauli, L. Lampe, and R. Schober, “‘Turbo DPSK’ using soft multiple-symbol differential sphere decoding,” IEEE Transactions on Information Theory, vol. 52, no. 4, pp. 1385–1398, 2006.

[24] S. ten Brink, F. Sanzi, and J. Speidel, “Two-dimensional iterative APP channel estimation and decoding for OFDM systems,” in Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM ’00), vol. 2, pp. 741–745, 2000.

[25] S. Y. Park, Y. G. Kim, C. G. Kang, and D. E. Kang, “Iterative receiver with joint detection and channel estimation for OFDM system with multiple receiver antennas in mobile radio channels,” in Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM ‘01), vol. 5, pp. 3085–3089, 2001.

[26] F. Sanzi and M. C. Necker, “Totally blind APP channel estimation with higher order modulation schemes,” in Proceedings of the 58th IEEE Vehicular Technology Conference (VTC ‘03), vol. 2, pp. 1167–1171, October 2003.

[27] T. May and H. Rohling, “Turbo decoding of convolutional codes in differentially modulated OFDM transmission systems,” in Proceedings of the 49th IEEE Vehicular Technology Conference (VTC ’99), vol. 3, pp. 1891–1895, July 1999.

[28] H. Rohling and T. May, “OFDM systems with differential modulation schemes and turbo decoding techniques,” in Proceedings of the International Zurich Seminar on Broadband Communications, pp. 251–255, 2000.

[29] P. Haase and H. Rohling, “Iterative detection of differentially modulated APSK signals in an OFDM transmission system,” in Proceedings of the The 5th International Symposium on Wireless Personal Multimedia Communications, vol. 2, pp. 711–714, October 2002.

[30] J. Hagenauer and P. Hoeher, “Viterbi algorithm with soft-decision outputs and its applications,” in Proceedings of the IEEE Global Telecommunications Conference and Exhibition. Communications Technology for the 1990s and Beyond (GLOBECOM ’89), vol. 3, pp. 1680–1686, 1989.

[31] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, “Optimal decoding of linear codes for minimizing symbol error rate (Corresp.),” IEEE Transactions on Information Theory, vol. 20, no. 2, pp. 284–287, 1974.

[32] W. J. van Houtum and F. M. J. Willems, “Joint and iterative detection and decoding of differentially encoded COFDM systems,” in Proceedings of the 17th IEEE International Conference on Telecommunications (ICT ‘10), pp. 36–43, April 2010.

[33] M. Franceschini, G. Ferrari, and R. Raheli, “Detection by multiple trellises,” IEEE Transactions on Communications, vol. 57, no. 3, pp. 726–737, 2009.

[34] J. Hagenauer, N. Seshadri, and C. E. W. Sundberg, “The performance of rate-compatible punctured convolutional codes for digital mobile radio,” IEEE Transactions on Communications, vol. 38, no. 7, pp. 966–980, 1990.

[35] J. G. Proakis and M. Salehi, Digital Communications, Higher Education Series, McGraw-Hill, 5th edition, 2008.

[36] “COST 207, digital land mobile radio communications,” Tech. Rep., Office for Official Publications of the European Communities, Luxembourg, UK, 1989.
Advanced Receiver Design for Mitigating Multiple RF Impairments in OFDM Systems: Algorithms and RF Measurements

Adnan Kiayani, Lauri Anttila, Yaning Zou, and Mikko Valkama

Department of Communications Engineering, Tampere University of Technology, 33101 Tampere, Finland

Correspondence should be addressed to Adnan Kiayani, adnan.kiayani@tut.fi

Received 15 July 2011; Accepted 12 October 2011

Academic Editor: Ming-Der Shieh

Copyright © 2012 Adnan Kiayani et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Direct-conversion architecture-based orthogonal frequency division multiplexing (OFDM) systems are troubled by impairments such as in-phase and quadrature-phase (I/Q) imbalance and carrier frequency offset (CFO). These impairments are unavoidable in any practical implementation and severely degrade the obtainable link performance. In this contribution, we study the joint impact of frequency-selective I/Q imbalance at both transmitter and receiver together with channel distortions and CFO error. Two estimation and compensation structures based on different pilot patterns are proposed for coping with such impairments. The first structure is based on preamble pilot pattern while the second one assumes a sparse pilot pattern. The proposed estimation/compensation structures are able to separate the individual impairments, which are then compensated in the reverse order of their appearance at the receiver. We present time-domain estimation and compensation algorithms for receiver I/Q imbalance and CFO and propose low-complexity algorithms for the compensation of channel distortions and transmitter IQ imbalance. The performance of the compensation algorithms is investigated with computer simulations as well as with practical radio frequency (RF) measurements. The performance results indicate that the proposed techniques provide close to the ideal performance both in simulations and measurements.

1. Introduction

With the ever-increasing demand for high data rates and high quality of services for end users, bandwidth-efficient transmission schemes such as orthogonal frequency division multiplexing (OFDM) are being adopted in emerging wireless communication systems (e.g., WLAN 802.11a/g/n [1], WiMAX IEEE 802.16 [2], DVB-T [3], DVB-H [4], 3GPP LTE [5]). The physical layer implementation of OFDM-based systems with direct-conversion (zero-IF or homodyne) radio architecture represents a promising solution for future wireless systems. The direct-conversion architecture offers a simplified analog front end (FE) as it performs the frequency translation in one step and thus eliminates the need of bulky image rejection filters [6, 7]. This yields an easy integration of analog and digital components of the FE on a single chip and consequently results in lower-cost and less power consumption. From the perspective of practical implementation, a trade-off exists between the high integrability and the performance. The direct-conversion architecture-based transceivers are extremely vulnerable to the nonidealities of analog front-end components. The main impairments that degrade the system performance are in-phase quadrature-phase (I/Q) imbalance, DC offset, and carrier frequency offset (CFO) [6, 7]. The adoption of higher-order modulation alphabets (such as 64-QAM) in OFDM systems suggests that they are increasingly sensitive to any impairments in the underlying analog hardware. Rather than trying to improve the quality of individual analog modules, it is more cost-efficient to tolerate these RF impairments to a certain degree in the analog domain and afterward compensating them in the digital domain.

The frequency up- and downconversion in the direct-conversion architectures are implemented by I/Q mixing, which suffers from the amplitude and phase mismatch between the I- and Q- branches [8–24]. This problem
is mainly inflicted by the modulators/demodulators which employ the principle of having equal gain and exact 90° phase difference between the quadrature branches. However, other analog front-end components such as DACs or ADCs and LPFs also contribute in general to the imbalance effects, resulting in frequency-selective I/Q imbalance [8–10]. Estimation and compensation schemes dealing with I/Q imbalance have been proposed in several papers including [8–12] and references therein.

CFO is another important RF impairment, particularly associated with OFDM-based communication systems. It is caused by the instability of local oscillator and also due to the mobility of users [25, 26]. OFDM systems divide the available bandwidth into many orthogonal subcarriers with a small subcarrier spacing. The subcarriers orthogonality is lost in the presence of CFO, leading to intercarrier interference. Because the CFO cannot be perfectly estimated in the presence of I/Q imbalances, most of the recent works (e.g., [18–21]) analyze the performance degradation and propose compensation scheme coping with CFO and receiver I/Q imbalance, but do not take into account the transmitter I/Q imbalance. Gil et al. [22], Chung and Phoong [23], and Tandur and Moonen [24] developed the joint estimation algorithm of the I/Q imbalance, CFO, and channel, however frequency-independent I/Q imbalance model is assumed. Finally, in [27], compensation methods for the mitigation of frequency-selective transmitter and receiver I/Q imbalance combined with CFO and channel distortions are proposed by the authors of this article. But the application of proposed algorithms is restricted to systems with a preamble pilot, and also the computational complexity for CFO estimation is very large.

In this paper, we consider DSP-based compensation of frequency-selective transmitter and receiver I/Q imbalance together with frequency-selective channel and CFO in the OFDM system context. The central theme is to first systematically formulate the baseband equivalent of the received signal. Based on this signal model, we propose new low-complexity decoupled estimation and compensation algorithms. Unlike the joint estimation algorithms (e.g., [22, 24]) where the compensator parameters have to be reestimated with channel variations, the decoupled schemes are advantageous in the sense that only the varying parameter has to be reestimated, while the other parameters remain the same. More precisely, at the estimation and compensation stage, we first compensate the generally time-invariant receiver I/Q imbalance in time domain with blind methods proposed in authors’ earlier work [11, 13]. The subsequent estimation and compensation of nonidealities including CFO, channel distortions, and transmitter IQ imbalance is done for pilot symbol-assisted modulation (PSAM) OFDM systems in each transmission. With practical considerations, two pilot patterns are taken into account in this work, namely, the preamble pilot pattern, where one complete OFDM symbol in a transmission frame is assumed known, and the sparse pilot pattern, where pilot symbols are placed over different subcarriers of certain OFDM symbols. We propose two alternative estimation and compensation structures depending on the pilot pattern. The leading principle in CFO estimation and compensation for both pilot patterns is to use an already existing technique of [28] and apply it in time domain. For the preamble pilot-based estimation, we propose zero-forcing (ZF) and maximum-likelihood (ML) estimation methods for joint channel distortions and transmitter I/Q imbalance compensation, whereas an algorithm for successive compensation of channel distortions and transmitter I/Q imbalance is proposed for sparse pilot-based structure. The performance of algorithms is evaluated with extensive computer simulations as well as with laboratory measurement setup.

The novelty of this paper is as follows:

1. We do not make any specific assumption about the location of pilots for sparse pilot-based estimation and compensation structure. The existing algorithms, for example [29], allocate the pilots to mirror frequency pairs which is generally not valid in practical radio systems like LTE [5].

2. In this work, instead of proposing comprehensive and brand new algorithms to cope with multiple RF impairments at once, we simplify the complexity of the problem by reforming overall receiver design and decoupling the effects of individual RF impairments. Then incorporating already existing efficient algorithms with new proposed methods, a hybrid time-and-frequency domain compensation architecture with very reasonable complexity and good performance is achieved.

3. Practical RF measurements are used to verify the applicability of algorithms in real-world receiver design which, to the best of authors’ knowledge, has not been addressed so far in the literature. The performance of individual impairments has been evaluated though, for example, [9, 11], but RF performance evaluation in the presence of all the considered impairments is still new.

4. Here we use two widely deployed pilot patterns in the current and upcoming OFDM-based radio systems (e.g., IEEE 802.11n [1], DVB-H/T [3, 4], and LTE [5] systems), namely, the preamble-based and sparsely located pilot structures for parameter estimations.

As a result, the proposed algorithms can be directly applied in the corresponding receiver design without any modification.

An attractive feature of receiver I/Q imbalance compensation algorithm is that it is able to track the time-variation of I/Q imbalance and updates the coefficients of compensation filter appropriately.

The paper is organized as follows. I/Q imbalance model and its impact, OFDM signal model under frequency-selective I/Q imbalances, channel distortions, and CFO are described in Section 2. Section 3 presents the estimation and compensation techniques for the mitigation of impairments. Computer simulation results are shown in Section 4. In Section 5, the measurement setup and obtained results are presented. Finally, the conclusions are drawn in Section 6.
Preliminaries. The notations used in this paper are as follows. Scalar parameters are represented by lower case letters $a$ and frequency domain quantities with upper case letters $A$. We denote the time domain vectors/matrices by lower case bold face with over-bar $\mathbf{a}/\mathbf{A}$ and frequency domain vectors/matrices by bold face letters $a/A$. Superscript $(\cdot)^{T}, (\cdot)^{*}$, and $(\cdot)^{-1}$ denote the transpose, conjugate, and inverse of a vector, a scalar, or a matrix, respectively. The $(i,j)$th quantity of a matrix is defined by $A_{ij}$ and $(i,i)$th quantity by $A_{i}$. The convolution operation is indicated by $\ast$. If $a = [a_{1}, a_{2}, \ldots, a_{N/2}, \ldots, a_{N}]^{T}$ is an OFDM symbol with $N$ subcarriers then $a^{*} = [a_{1}^{*}, a_{2}^{*}, \ldots, a_{N/2}^{*}, \ldots, a_{N}^{*}]^{T}$ is the conjugate of mirror subcarriers vector. The discrete Fourier transform operation is denoted by a matrix $F$ of size $N \times N$ defined as $[F]_{mn} = e^{-j2\pi mn/N}, m = 0, 1, \ldots, N - 1; n = 0, 1, \ldots, N - 1$.

2. Signal and System Model

2.1. Frequency-Selective I/Q Imbalance Model. The direct-conversion transmitter architecture is based on the principle of directly I/Q upconverting the baseband signal to the RF frequency. The upconversion is performed in the analog domain by a quadrature mixer, which theoretically provides infinite image signal attenuation. This eliminates the need for image rejection filter, relaxing the overall requirement for RF filtering. A perfectly balanced modulator corresponds to equal gain and $90^\circ$ phase difference between the quadrature branches. However, in practice, this requirement is not fully satisfied. In addition to that, other modem components such as DACs and LPFs in the I- and Q-branch are not perfectly matched [7, 8]. These effects are called I/Q imbalance, and it results in the limited suppression of the image signal [8, 9]. In the wideband system context, the reconstruction filters in the modulator exhibit frequency-dependent response, which causes the I/Q imbalance to vary as a function of frequency. Consequently, we can characterize the I/Q imbalance with transmitter gain imbalance and phase imbalance parameter $g_T$ and $\phi_T$, respectively [8, 27]. In addition, the relative nonideal filter transfer function between in the I- and Q-branches is modeled with the filter $h_T(t)$. A conceptual block diagram is illustrated in Figure 1. To see the impact of I/Q imbalance on the transmitted signal, we denote the baseband equivalent ideal transmit signal as $z(t) = z_I(t) + jz_Q(t)$. The complex envelope of the transmitted RF signal with I/Q imbalance effects is [8]

$$s(t) = g_{1,R}(t) \ast z(t) + g_{2,R}(t) \ast z^{*}(t), \quad (1)$$

where the complex filters $g_{1,R}(t)$ and $g_{2,R}(t)$ correspond to imbalance filters and are defined as

$$g_{1,R}(t) = \frac{\delta(t) + g_{R} \exp(-j\phi_{R})h_{R}(t)}{2},$$
$$g_{2,R}(t) = \frac{\delta(t) - g_{R} \exp(j\phi_{R})h_{R}(t)}{2}. \quad (2)$$

The baseband equivalent signal in frequency domain is

$$S(f) = G_{1,R}(f)Z(f) + G_{2,R}(f)Z^{*}(-f). \quad (3)$$

The above equation indicates that the imbalanced baseband signal is weighted sum of the desired signal $Z(f)$ and the undesired image signal $Z^{*}(-f)$. The undesired image signal term is produced by the I/Q imbalance and results in the mirror frequency interference.

On the receiver side, the RF signal is downconverted to baseband using a quadrature demodulator. The downconverted baseband equivalent of the received signal $r(t)$ is of the form [11]

$$x(t) = g_{1,R}(t) \ast r(t) + g_{2,R}(t) \ast r^{*}(t) \quad (4)$$

with the receiver I/Q imbalance filters $g_{1,R}(t)$ and $g_{2,R}(t)$ given as

$$g_{1,R}(t) = \frac{\delta(t) + g_{R} \exp(-j\phi_{R})h_{R}(t)}{2},$$
$$g_{2,R}(t) = \frac{\delta(t) - g_{R} \exp(j\phi_{R})h_{R}(t)}{2}. \quad (5)$$

The Fourier transform of (4) is

$$X(f) = G_{1,R}(f)R(f) + G_{2,R}(f)R^{*}(-f). \quad (6)$$

Again the mirror-frequency interference due to I/Q imbalance is evident from (6).

The image rejection ratio (IRR) quantifies the image signal suppression and is defined as the ratio of desired signal power to image signal power, expressed in dB as

$$\text{IRR}_{T,R}(f) \triangleq 10\log_{10}\left(\frac{|G_{1,R}(f)|^{2}}{|G_{2,R}(f)|^{2}}\right)\text{[dB].} \quad (7)$$

With careful analog design, IRR’s in the order of 25–40 dB are stated feasible [6–8].

2.2. OFDM Signal Model with RF Impairments. We consider an OFDM system with $N$ subcarriers. At the transmitter, OFDM symbols are generated by computing the $N$-point inverse discrete Fourier transform (IDFT) of the data symbols $u$. A cyclic prefix (CP) of length $L_{CP}$ is appended ahead of each OFDM symbol which is removed at the receiver after DFT.
Referring to Figure 2, consider the transmission of the $i$th symbol of size $N \times 1$, the frequency domain transmitted signal corresponding to the $i$th OFDM symbol is then given by

$$s^i = G_{1,i}u^i + G_{2,i}u^i\theta,$$  

(8)

where $G_{1,i}$ and $G_{2,i}$ are $N \times N$ frequency domain diagonal matrices whose diagonal elements possess frequency responses of transmitter I/Q imbalance filters ($g_{1,i}; g_{2,i}$ in (2)). We consider the transmission over a block fading channel whose impulse response is denoted by $h = [h(0), h(1), \ldots, h(L-1)]^T$ with channel length $L$. The received frequency domain signal can then be expressed as

$$r^i = H_{1,i}u^i + H_{2,i}u^i\theta + n,$$  

(9)

where $H$ is $N \times N$ frequency domain matrix that contains the channel frequency response on its main diagonal and $n$ is additive white Gaussian noise (AWGN).

Assume now that the local oscillator of transmitter and receiver are not synchronized and a CFO $\Delta f$ is present. If we denote the time domain received signal vector as $\mathbf{r}^i$, then the signal in the presence of CFO can be modeled as $\mathbf{r}^i = \mathbf{\Omega}^i\mathbf{r}^i$, where $\mathbf{\Omega}^i = \text{diag}([\mathbf{c}])$ is the CFO matrix with main diagonal entries

$$c^i = e^{j2\pi fT_s(i-1)N+iL_{CI}} \times \left[1, e^{j2\pi fT_s}, \ldots, e^{j2\pi fT_s(N-1)}\right]^T.$$  

(10)

Here $\Delta f$ is the frequency offset in Hertz and $T_s$ is the sampling rate. The frequency domain $N \times N$ CFO matrix $\mathbf{\Omega}^i = \mathbf{\Omega}^i\mathbf{1}$ is circulant in nature and has $c(0)$ on its main diagonal and $c(1)$ and $c(N-1)$ on the first sub- and super-diagonal [30]. Including the CFO effect to (9), we obtain

$$r^i = \mathbf{\Omega}^i H_{1,i}u^i + \mathbf{\Omega}^i H_{2,i}u^i\theta + n.$$  

(11)

Finally taking into account the receiver I/Q imbalance, the signal at the downconversion stage appears as

$$x^i = G_{1,R}r^i + G_{2,R}r^i\theta,$$  

(12)

where again $G_{1,R}$ and $G_{2,R}$ are $N \times N$ frequency domain diagonal matrices whose diagonal elements are DFT of receiver I/Q imbalance filters ($g_{1,R}; g_{2,R}$ in (5)). After some simple mathematical manipulations, the complex baseband equivalent of the received signal in terms of transmitter/receiver I/Q imbalance, channel, and CFO is given by

$$x^i = \left( G_{1,R} \Omega^i H_{1,R} + G_{2,R} \Omega^i H^i H_{2,R}^i \right) u^i + \left( G_{1,R} \Omega^i H_{2,R} + G_{2,R} \Omega^i H^i H_{1,R}^i \right) u^i\theta + v,$$  

(13)

where $\Omega^i$, $H^i$, $G_{1,i}$, $G_{2,i}$ are the mirrored matrix of $\Omega^i$, $H$, $G_{1,i}$, $G_{2,i}$, respectively. The noise term now becomes $v = G_{1,R}n + G_{2,R}n\theta$, whose elements are still complex circular Gaussian, but with correlated mirror subcarriers. In the ideal case when there is perfect transmitter/receiver modulator/demodulator matching as well as perfect synchronization between the transmitter/receiver local oscillator, the model in (13) reduces to $x^i = Hu^i + n$ which can be equalized with one complex multiplication for each subcarrier in each OFDM symbol. As evident from (13), both I/Q imbalances and CFO introduce ICI and severely degrade the system performance, thus digital compensation is needed.

In this paper, we assume that the length of the cascade of impulse responses of the transmitter I/Q imbalance filters ($g_{1,i}(t)$ and $g_{2,i}(t)$ in (1)), the multipath radio channel, and the receiver I/Q imbalance filters ($g_{1,R}(t)$ and $g_{2,R}(t)$ in (4)) does not exceed the length of CP. It is reasonable to make such an assumption as the frequency-selective transceiver I/Q imbalance is rather moderate and can be estimated with sufficient accuracy with only a few taps [8, 11, 27].

### 3. Estimation and Compensation of Nonidealities

The estimation and compensation of radio frequency impairments in OFDM systems can be performed blindly and/or with the aid of training (a.k.a. pilot) symbols. For forming such pilot-symbol-assisted OFDM systems, it involves inserting the known symbols in the stream of data symbols. With practical system design, two types of pilot patterns are widely used—the preamble pilot pattern that amends an entirely known OFDM symbol at the beginning of the frame and the sparse pilot pattern where pilot symbols are sparsely inserted at some subcarriers of specific OFDM symbols. Figure 3 illustrate these two ways of inserting pilots among the data symbols.

In the following, we discuss receiver-based estimation and compensation algorithms for mitigating all the considered RF impairments. With developed signal model and given pilot patterns, we are able to isolate them individually and then process them for estimation and compensation in the reverse order of their appearance in the transmitter and receiver front end. We first discuss the receiver I/Q imbalance estimation and compensation, which is performed in time domain and is independent of the pilot pattern. Then, two low-complexity and novel structures are proposed for CFO, channel, and transmitter I/Q imbalance compensation with both preamble and sparse pilot structures. The CFO
3.1. Blind Receiver I/Q Imbalance Compensation. For receiver I/Q imbalance estimation and compensation, we propose to utilize the statistical signal processing-based blind I/Q imbalance compensation algorithms, developed in authors’ earlier work [11, 13, 14]. In general, most complex communication systems utilize the statistical signal processing-based blind I/Q imbalance estimation and compensation, we propose to apply the compensation structure proposed in this section to the received signal (without receiver I/Q imbalance). The compensation structure of [11] is to recover the circularity using second-order statistics-based approach, under the assumption that the received signal (without receiver I/Q imbalance) is circular. The circularity of received signal is inherently lost in the presence of transmitter I/Q imbalance, potentially hindering the applicability of the I/Q imbalance compensation algorithms of [11]. However, it is shown in [13] that either a CFO or a fading channel will restore the circularity of the received signal (without receiver I/Q imbalance), thus making it possible to utilize the circularity-based algorithms of [11]. These blind algorithms are also beneficial from the point of view that they do not affect the subsequent signal processing of CFO, channel, and transmitter I/Q imbalance estimation and compensation.

For notational simplicity, we drop the noise term in (13) and continue the analysis. Notice that even though the actual compensation processing is done partly in the time domain (for receiver I/Q imbalance and CFO), the following analysis is done completely in the frequency domain. From (12), we recognize that the signal \( x' \) is the linear combination of the received signal \( r' \) and its mirror conjugate \( r'^* \). For such a system model, the natural compensator is of the form

\[
y'(i) = x'(i) + W_R x'^*(i). \tag{14}
\]

Substituting the observed signal \( x' \) of (12) into the compensator in (14), the output can be written as

\[
y(i) = \left( G_{1,R} + W_R G_{2,R}^* \right) x'(i) + \left( G_{2,R} + W_R G_{1,R}^* \right) r'^*(i). \tag{15}
\]

From (15), it follows that the optimum compensation filter \( W_R \) canceling the mirror-frequency interference is, in frequency domain \( N \times N \) diagonal matrix, of the form \( W_{OPT} = -G_{2,R}(G_{1,R}^*)^{-1} \) and is derived in [11]. Applying the optimum compensation filter to (15) cancels the mirror conjugate term \( r'^* \) and results in the output signal as

\[
y(i) = \left( G_{1,R} \Omega^T H G_{1,T} - G_{2,R} G_{2,R}^* (G_{1,R}^*)^{-1} \Omega^T H G_{1,T} \right) u(i) \\
+ \left( G_{1,R} \Omega^T H G_{2,T} - G_{2,R} G_{2,R}^* (G_{1,R}^*)^{-1} \Omega^T H G_{1,T} \right) u'^*. \tag{16}
\]

It is well known that \( |G_{1,T,R}| > |G_{2,T,R}| \) and \( |G_{1,T,R}| \approx 1 \) for any practical imbalance values; thus the contributions of the second terms inside the parentheses are relatively very small compared to the first term and can be omitted, leading to the following form:

\[
y(i) \approx G_{1,R} \Omega^T H \left( G_{1,T} u(i) + G_{2,T} u'^* \right). \tag{17}
\]

Comparing (16) with (13) suggests that, given \( W_{OPT} \), the compensator is able to fully suppress the image signal term generated by a frequency-selective receiver I/Q imbalance in the presence of CFO, channel distortions, and transmitter I/Q imbalance. The estimation of \( W_{OPT} \) is described in [11, 13] and is not reproduced here. Next, we discuss the pilot-based estimation and compensation schemes for CFO, channel distortions, and transmitter I/Q imbalance.

3.2. Estimation and Compensation with Preamble Pilot Structure. The compensation structure proposed in this section...
3.2.1. CFO Estimation and Compensation. There exists an abundant literature on CFO estimation (see, e.g., [18–25] and references therein), but most of these algorithms assume a specific pilot structure which restrict their use in our problem. The subspace-based CFO estimation approach proposed in [26] utilizes only one OFDM symbol, and can be directly used here at the expense of increased computational complexity. We rather suggest to use the CFO estimation algorithm of [28] as discussed in [25]. More specifically, the actual CFO estimation is performed in two stages: first the fractional part of CFO is estimated and corrected in time domain by using the CP correlation. The integer part is then estimated in frequency domain in the second stage. Even though the presence of transmitter I/Q imbalance is not taken into account in this method, we show with simulations that the algorithm is still able to achieve reasonable good CFO estimation performance.

Assuming perfect CFO estimate is obtained, we perform CFO compensation in time domain by multiplying the time domain equivalent of (17) with the complex conjugate of $\hat{\Omega}^\dagger$, which, in time domain, can be written as

$$\hat{\mathbf{y}} = \mathbf{F}[\hat{\Omega}^\dagger \mathbf{y}] = \hat{\mathbf{G}}_{1,R} \mathbf{H} (\mathbf{G}_{1,T} \mathbf{u}^i + \mathbf{G}_{2,T} \mathbf{u}^\text{ibm})^\dagger,$$

where $\hat{\mathbf{G}}_{1,R} \triangleq \Omega^\dagger \mathbf{G}_{1,R}$.

3.2.2. Joint Transmitter I/Q Imbalance and Channel Equalization. The joint transmitter I/Q imbalance and channel equalization are carried out in frequency domain. To simplify the analysis further, let us define the direct and image signal filters as $\mathbf{G}_D \triangleq \hat{\mathbf{G}}_{1,R} \mathbf{H} g_{1,T}$ and $\mathbf{G}_M \triangleq \hat{\mathbf{G}}_{1,R} \mathbf{H} g_{2,T}$, which transforms (18) into

$$\hat{\mathbf{y}} = \mathbf{G}_D \mathbf{u}^i + \mathbf{G}_M \mathbf{u}^\text{ibm}.$$

We now switch to the subcarrier model and write the CFO compensated symbol for the $k$th subcarrier and the complex conjugate of its mirror subcarrier. In the subsequent derivations, the subcarrier index $-k$ refers to the physically opposite mirror subcarrier

$$\begin{bmatrix} \hat{y}_k^i \\ \hat{y}_{k,-k}^i \end{bmatrix} = \begin{bmatrix} G_{D,k} & G_{M,k} \\ G_{M,-k}^* & G_{D,-k}^* \end{bmatrix} \begin{bmatrix} U_k^i \\ U_{-k}^i \end{bmatrix}.$$

The matrix $G_{\text{total},k}$ represents the joint transmitter I/Q imbalance and channel response. Assuming the coefficients of matrix $G_{\text{total},k}$ are known, we are able to directly apply the ZF or the ML detection principle on the mirror subcarrier pair to estimate the transmitted symbol.

With ZF equalization scheme, the estimate of original transmit symbols is obtained by solving (20) for each subcarrier $k$ and its mirror subcarrier $-k$ as

$$\begin{bmatrix} \hat{U}_k^i \\ \hat{U}_{-k}^i \end{bmatrix} = \hat{G}_{\text{total},k}^{-1} \begin{bmatrix} \hat{y}_k^i \\ \hat{y}_{k,-k}^i \end{bmatrix}.$$

It is well known that ZF equalization suffers from the noise enhancement problem, yet it gives good calibration performance as will be shown in Section 4.

Alternatively, the transmitted symbols can be estimated using ML detection principle which is based on the principle of minimizing the cost function

$$\begin{bmatrix} \hat{U}_k^i \\ \hat{U}_{-k}^i \end{bmatrix} = \arg\min_{U_k^i, U_{-k}^i} \| \hat{G}_{\text{total},k} \begin{bmatrix} \hat{U}_k^i \\ \hat{U}_{-k}^i \end{bmatrix} - \begin{bmatrix} \hat{y}_k^i \\ \hat{y}_{k,-k}^i \end{bmatrix} \|^2,$$

where $U_k^i$ in (22) denotes the trial value of OFDM symbol $U_k^i$. The ML equalizer is able to exploit the frequency diversity induced by transmitter I/Q imbalance and gives better performance than ZF, but the computational complexity associated with ML is very large which increases with higher-order constellations and large number of subcarriers.
3.2.3. LS Identification of $G_{\text{tot},K}$. We consider the LS estimation of the joint transmitter I/Q imbalance and channel filters $G_D$ and $G_M$. The filters are estimated in time domain similar to [32], where a time-domain ML channel estimator was proposed for OFDM. This approach can utilize the frequency domain correlation of the channel implicitly without making any statistical assumptions about the channel, through limiting the length of the estimated channel response estimates as $N_{\hat{g}} \ll N$.

We switch to matrix-vector algebra and write the time-domain received symbol corresponding to preamble pilot as $\tilde{y}_{p}(n) \triangleq [\tilde{y}_{p}(n), \tilde{y}_{p}(n-1), \ldots, \tilde{y}_{p}(n-N+1)]^T$ and the observable imbalance filters as $\hat{g}_{D,M} = [\hat{g}_{D,M,1}, \hat{g}_{D,M,2}, \ldots, \hat{g}_{D,M,N}]^T$, where $N$ is the length of pilot data and $N_{\hat{g}} < L_{CP}$ is the length of estimated filters. The received symbol vector during the pilot symbol is then

$$\tilde{y}(n) = U_b^T(n) \hat{g}_D + U_b^P(n) \hat{g}_M = \begin{bmatrix} U_b^T(n) \\ U_b^P(n) \end{bmatrix} \begin{bmatrix} \hat{g}_D \\ \hat{g}_M \end{bmatrix},$$

(23)

where $U_b(n)$ is the time-domain circular-convolution matrix formed from IDFT of pilot symbol $u_p$. The LS estimates of imbalance filters can be computed as [33]

$$\begin{bmatrix} \hat{g}_D \\ \hat{g}_M \end{bmatrix} = U_b^P(n)^T \tilde{y}(n).$$

(24)

Here $U_b^P(n)^T$ represents the pseudoinverse of $U_b^P(n)$ and is given by $U_b^P(n)^T = (U_b^P(n) U_b^P(n)^T)^{-1} U_b^P(n)^T$ [33]. Frequency domain expressions of the estimated filters $G_D$ and $G_M$, which are needed for equalization/detection in (20)–(22), are then obtained through DFT.

3.3. Estimation and Compensation with Sparse Pilot Structure. OFDM systems such as LTE and DVB-T/H do not include a preamble pilot in their frame, rather the pilot tones are inserted sparsely in the OFDM symbols. DVB-T/H standard defines both continual and scattered pilots, on the other hand, LTE includes pilot subcarriers only on specified OFDM symbols. Figure 5 shows the reference pilot structure for LTE system with pilot symbols at every sixth subcarrier during the first and fifth OFDM symbol of each slot.

One big challenge with sparse located pilot structure is that the pilots are most likely not allocated to mirror-frequency pairs which is required by most of the pilot-based algorithms developed in literature, for example, [29]. The compensation algorithms proposed in this section consist of estimating the impairments at the pilot frequencies and then interpolating the estimates over all the subcarriers. Therefore, no pilot structure modification is needed, and they can be directly applied for the development of, for example, LTE receiver. The estimation and compensation structure is illustrated in Figure 6. Again, notice that the receiver I/Q imbalance has already been compensated with the blind method discussed in Section 3.1.

3.3.1. CFO Estimation and Compensation. For the CFO estimation, we again apply the two-step time-domain approach of [25] as discussed in the previous subsection, resulting in the time domain CFO compensated signal similar to (18) as

$$\tilde{y} = \hat{G}_{1,r} H_{G,1,T} u + \hat{G}_{1,r} H_{G,2,T} u^* = G_D u + G_M u^*.$$

(25)

3.3.2. Decoupled Transmitter I/Q Imbalance and Channel Equalization. With the sparse pilot structure, we propose to compensate the channel and transmitter I/Q imbalance successively. Algorithm 1 summarizes the algorithm for the estimation of transmitted symbols.

**Algorithm 1** (It is for estimation of channel distortions and transmitter I/Q imbalance).

1. Estimate widely-linear filter $G_{D,k}$ (see Section 3.3.3).
2. Switch to subcarrier model and write (25) in frequency domain for subcarrier $k$ as

$$\hat{Y}_k = G_{D,k} U_k + G_{M,k} U_k^*.$$

(26)

3. Divide (26) by $G_{D,k}$ to obtain the equalized subcarriers as

$$\hat{S}_k = \frac{1}{G_{D,k}} \hat{Y}_k = U_k + \frac{G_{M,k}}{G_{D,k}} U_k^*$$

$$= U_k + \frac{G_{D,k} U_k^*}{G_{M,k}}.$$

(27)

Clearly, the channel effect is removed and the only impairment present in the signal now is the transmitter I/Q imbalance.

4. Hard-decision-based detection of the mirror subcarriers of equalized symbol, given by

$$\tilde{S}_k = D(\hat{S}_k),$$

(28)

where $D$ denotes the decision.

5. At the pilot subcarriers, we now have

$$\tilde{S}_k = U_k^p - W_k^p \hat{S}_k.$$

(29)
From above equation, we find the compensation filter at the pilot subcarriers as

\[ W_k^p = \frac{U_k^p - S_k^p}{S_{k}^p} \] (30)

(7) Interpolate \( W_k^p \) to all subcarriers to obtain \( W_k \) (see Section 3.3.4).

(8) The estimated symbols are then given as

\[ \hat{y}_k^i = S_k^i + W_k S_{k}^i. \] (31)

3.3.3. Estimation of Widely Linear Filter \( G_{D,K} \). The filter \( G_{D,K} \) can also be estimated in the time domain as discussed in the previous subsection. For the \( i \)-th received OFDM symbol after receiver I/Q imbalance and CFO compensation, we perform the DFT operation and pick the data on pilot subcarriers. The time domain OFDM symbol corresponding to pilot subcarriers can then be written as \( \bar{Y}(n) \triangleq [\hat{y}_1(n), \hat{y}_1(n-1), \ldots, \hat{y}_1(n-N_p + 1)]^T \) and the observable imbalance filter as \( \bar{g}_D = [g_{D,1}, g_{D,2}, \ldots, g_{D,N_p}]^T \), where again \( N_p \) is the length of sparse pilot symbols and \( N_k < L_{CP} \) is the length of estimated filters. The LS estimate of \( \bar{g}_D \) is then

\[ \bar{g}_D = \left( \bar{U}_p(n)^H \bar{U}_p(n) \right)^{-1} \bar{U}_p(n)^H \bar{y}(n). \] (32)

where \( \bar{U}_p(n) \) is the time-domain circular-convolution matrix formed from by taking the IDFT of pilot symbols \( u_p \).

3.3.4. Interpolation of \( W_k^p \). Here, we derive the transformation for the interpolation of filter \( W_k^p \) in (30). Assume that the pilot estimates stem from a time domain impulse response \( \bar{w} \) of length \( N_w \), then the model expressed in (30) becomes

\[ \bar{W}^p = \bar{F}_{(\text{indP},1:1)} \bar{w}, \] (33)

where \( \bar{W}^p \) is \( N_p \times N_p \) diagonal matrix and \( \text{indP} \) denotes the indices of pilot subcarriers. Now, let us define the DFT matrix \( \bar{F}_M \) comprised of first \( N_p \) columns and rows corresponding to the pilot subcarriers, formally given as

\[ \bar{F}_M = \bar{F}_{(\text{indP},1:1)}. \] (34)

then the impulse response vector is estimated based on minimizing the mean square error

\[ \arg \min_{\bar{w}} \| \bar{W}^p - \bar{F}_M \bar{w} \|^2 \] (35)

which can be found by minimizing the LS scheme, corresponding to left multiplying the pseudo-inverse of \( \bar{F}_M \) to (30) yielding

\[ \bar{w}_{LS} = \left( \bar{F}_M^H \bar{F}_M \right)^{-1} \bar{F}_M^H \bar{W}^p. \] (36)

The interpolated frequency-domain coefficients are then obtained as

\[ \bar{W}_{LS} = \bar{F}_{(1:N,1:N)} \bar{w}_{LS}. \] (37)

The estimates obtained from the above equation are then plugged in (31) to obtain transmitted symbols estimate.

In practice, some of the coefficients of \( \bar{W}_{LS} \) might be very biased due to deep channel fading. These biased estimates can be improved by using weighted-least-squares (WLS) solution. There are many possibilities for defining the weight matrix, for example, giving more weight to the subcarriers that have strong mirror subcarrier. In this work, we define the weight matrix as

\[ P_k = \text{diag} \left\{ \frac{|\hat{U}_k|^2}{|\hat{U}_k|^2} \right\}, \] (38)

which transform the original LS problem into WLS as

\[ \bar{W}_{WLS} = \left( \bar{F}_M^H P \bar{F}_M \right)^{-1} \bar{F}_M^H \bar{W}^p \] (39)

or equivalently in frequency domain as

\[ \bar{W}_{WLS} = \bar{F}_{(1:N,1:N)} \bar{w}_{WLS}. \] (40)

3.4. Compensation Complexity. In Table 1, we summarize the complexity in terms of number of the complex multiplications required by each compensation stage.

The overall complexity of the proposed compensation schemes is \( (N_w+1) \times (N + L_{CP}) + 2N_g \).
4. Simulation Results

The performance of proposed algorithms is illustrated in this section with computer simulations. We first evaluate the performance of CFO estimator in the presence of transmitter and receiver I/Q imbalance and channel distortions and do not take into account receiver I/Q imbalance, for both preamble and sparse pilot structures. Then, SER simulations are performed to evaluate the detection performance of compensation schemes in the presence of transmitter and receiver I/Q imbalance, channel distortions, and CFO.

4.1. Performance of CFO Estimator. The parameters considered for CFO estimator simulations are as follows: OFDM-based system with 64-QAM subcarrier modulation, total number of subcarriers \( N = 1024 \), of which 600 are active, 50 OFDM symbols, and the subcarrier spacing is 15 KHz. In the case of preamble pilot, the first complete OFDM symbol is considered as pilot and used for the CFO estimation. For sparse pilot example, pilot symbols are inserted at every sixth subcarrier. The multipath channel is 39 taps long with maximum delay spread of 2.5 ms and power delay profile as described in [34]. The transmitter I/Q imbalance filters impulse responses are \( \mathbf{h}_{x,T} = [1, 0, 0]^T \); \( \mathbf{h}_{g,T} = [0.97, -0.1, 0.01]^T \). Frequency-independent gain \( g_T \) and phase \( \phi_T \) are chosen corresponding to IRR levels of 20 dB, 30 dB, 40 dB, and infinite (INF.). These IRR levels map to gain imbalance of 10%, 3%, 1%, 0% and phase imbalance of 10°, 4°, 1°, 0°, respectively. The mean received SNR is 20 dB. The performance results are illustrated by plotting the mean square error (MSE) over normalized CFO values with 500 independent channel realizations: \( \text{MSE} = E[|\Delta f - \Delta f|^2] \).

Figures 7 and 8 show the performance of CFO estimation scheme of [25] for preamble and sparse pilot cases, respectively. For the sparse pilot example, we assume that the system under consideration is LTE for which normalized CFO can be higher than one [25], and we simulate the algorithm over the normalized CFO range of \([-5, 5]\). As evident from the results, the algorithm delivers sufficiently reliable estimates for small to large imbalance values. The MSE remains below than \( 10^{-7} \) with preamble and in the order of \( 10^{-5} \) when pilots are inserted sparsely at SNR = 20 dB.

4.2. SER Performance. Now, we perform the SER simulations and illustrate the results by plotting the mean SER as a function of received SNR. A typical 64-QAM OFDM system is considered whose parameters are given in Table 2. The OFDM signal parameters are similar to 3GPP long-term evolution (LTE) specifications [35]. The I/Q imbalance
Table 2: System parameters.

| Parameter                        | Value       |
|----------------------------------|-------------|
| Total number of subcarriers      | $N = 1024$  |
| Active subcarriers               | $N_a = 600$ |
| Cyclic prefix length             | $L_{CP} = 52$ |
| Subcarrier spacing               | $1/NTs = 15$ KHz |
| Sampling Frequency               | $F_s = 15.36$ MHz |
| Channel type                     | vehicular A |
| No. of channel taps              | $39$       |
| Gain imbalance                   | $g_{T,R} = 5\%$ |
| Phase imbalance                  | $\phi_{T,R} = 5^\circ$ |
| I-branch filter                  | $\mathbf{h}_{I,T,R} = [1, 0, 0]^T$ |
| Q-branch filter                  | $\mathbf{h}_{Q,T,R} = [0.97, -0.05, 0.01]^T$ |

values stated in the table correspond to 23–30 dB front-end IRR and represent a practical example case.

In the following figures, legend “No FE Distortion” refers to the case when no transmitter and receiver I/Q imbalance and CFO is present and known channel estimates are used for channel equalization, “W/o Compensation” refers to when all radio impairments are present and only channel is equalized with known estimates, and “W/Tx Rx IQ Imbalance only” for the case when only transmitter and receiver I/Q imbalance is present and channel is equalized with known estimates. “W/ZF Compensation” “W/ML Compensation” “W/LS Compensation”, and “W/WLS Compensation” legends exemplify the schemes proposed in Section 3. The results depicted are averaged over $10^3$ independent channel realizations.

For preamble pilot-based compensation, we transmit 50 OFDM symbols and use first OFDM symbol as pilot. In each simulation run, we generate a random CFO in the range $[-1, 1]$ and introduce it to the signal (with transmitter I/Q imbalance and channel distortions). The receiver I/Q imbalance is compensated with 3-tap compensation filter, built by utilizing the second-order statistics of the received signal. This is followed by CFO estimation and compensation, performed in time domain. ZF and ML estimation techniques are applied next. It can be seen in Figure 9 that communication system becomes completely unusable when no compensation scheme is in place and SER remains very high when only I/Q imbalance is present at both transmitter and receiver. The ZF equalizer is able to reduce the SER to the level of a system with no front-end distortion. On the other hand, the ML equalizer even outperforms the SER performance of system with no RF impairments and known channel estimates. In the low SNR region, the SER is close to the ideal system but as the SNR increases, the equalizer is able to extract the frequency diversity gain induced by transmitter I/Q imbalance. A similar diversity effect is also reported in [29] where only transmitter I/Q imbalance and channel distortions are considered.

The OFDM signal model for sparse pilot-based compensation is similar to LTE frame structure shown in Figure 10, consisting of one frame that is composed of ten subframes; each subframe is further divided into two slot. A slot consists of seven OFDM symbols, and pilot symbols are located at every sixth subcarrier of first and fifth OFDM symbol of each slot. Thus, in total, we transmit 140 OFDM symbols with pilot tones on 20 symbols. As discussed in the previous subsection, the normalized frequency offset is randomly generated in the range $[-5, 5]$ during each channel realization (CFO in LTE is much smaller, in fact it can be greater than one), and its effect is added to the signal. At the algorithm compensation level, we again apply a 3-tap compensation filter for receiver I/Q compensation, followed by CFO estimation and compensation. The LS and WLS compensation algorithms are applied in frequency
5. Experimental RF Measurements Results

5.1. Prototype Implementation. The generic structure of the measurement setup is identical to the one shown in Figures 4 and 6, and the corresponding block diagram is depicted in Figure 11. The prototype implementation integrates Rohde and Schwarz (R&S) AFQ100A baseband I/Q signal generator, R&S FSG spectrum and signal analyzer, R&S SMJ vector signal generator, HP E4422B RF signal generator, LeCroy 534AL quad-channel oscilloscope, Analog Devices (AD) AD8349 direct conversion modulator evaluation kit, and Maxim 2023 demodulator evaluation kit. All the instruments are connected to a computer via general purpose interface bus (GPIB) for remote control operation.

5.2. Measurement Setup. A computer with MATLAB is used for performing digital signal processing-related tasks and also for instrument control. The baseband OFDM signal with parameters similar to simulations as in Section 4 is generated in computer, and its samples are loaded into the memory of R&S AFQ. The analog I/Q signal output by R&S AFQ is upconverted to RF frequency by AD8349 modulator. In order to introduce the channel distortions and noise, we capture the baseband equivalent of the RF signal using built-in I/Q subsystem of R&S FSG. The I/Q subsystem first translates the signal to an intermediate frequency (IF) of 20.4 MHz, followed by IF filtering. The filtered signal is then digitized by a built-in ADC at a rate of 81.6 MHz. Finally the sampled signal is digitally downconverted to complex baseband, low-pass filtered, and resampled to desired sampling frequency. The samples are then read out in block of 512,000 each in ASCII format. The channel distortions and noise are then added, and the samples of signal (with transmitter I/Q imbalance, channel distortions, and noise) are loaded into R&S SMJ memory, which outputs the RF signal. The CFO effect is introduced at the receiver side by drifting the LO frequency of R&S SMJ relative to the RF frequency of HP E4422B. The LO frequency is used in the measurements in 1.5 GHz. On the receiver side, MAX2023 demodulator downconverts the signal from RF to baseband.
The output of analog frontend of MAX2023 is captured with a quad channel oscilloscope. The samples recorded by the oscilloscope are then sent to MATLAB for offline processing.

5.3. Performance Analysis

5.3.1. Front-End IRR without Calibration. We first shortly address the estimation of front-end IRR of AD8349 and MAX2023 chips, without any compensation. Although the exact values of gain and phase imbalance of these chips are unknown, the IRR analysis is still possible using LS fitting approach as follows. We denote the reference time domain signal \( s(n) \) to be transmitted. The complex envelope of RF signal can be written as in (41). \( Z(n) \) and \( Z^*(n) \) are time domain convolution matrix and conjugate convolution matrix formed from reference signal \( Z(n) \) and its conjugate \( Z^*(n) \), respectively. Assuming perfect synchronization in time, frequency, and phase, the LS estimates of the front-end IRR filters are then given by (42) [8, 11].

In Figure 12, IRR curves with different I/Q imbalance filter lengths, utilizing 50,000 samples and ensemble averaged over ten independent simulation runs, are plotted. The LO frequency is 1.5 GHz, and the signal is a 16-QAM modulated signal with 18 MHz two-sided bandwidth, six times oversampling, and root raised cosine pulse shaping. The results suggest that both chips have frequency selective image rejection, varying between 27–29 dB for AD8349 (transmitter modulator) and 30–33 dB for Maxim 2023 (receiver demodulator):

\[
\mathbf{s}(n) = \begin{bmatrix}
    s(n) \\
    s(n-1) \\
    \vdots \\
    s(n-L+1)
\end{bmatrix}
\]

\[
= \begin{bmatrix}
    g_1^T(n)Z(n) + g_2^T(n)Z^*(n) \\
    g_1^T(n)Z(n-1) + g_2^T(n)Z^*(n-1) \\
    \vdots \\
    g_1^T(n)Z(n-L+1) + g_2^T(n)Z^*(n-L+1)
\end{bmatrix}
\]

\[
\begin{bmatrix}
    g_{1,1}(n) \\
    g_{1,2}(n) \\
    \vdots \\
    g_{1,T}(n)
\end{bmatrix}
\begin{bmatrix}
    \mathbf{Z}_{1,1}(n) \\
    \mathbf{Z}_{1,2}(n) \\
    \vdots \\
    \mathbf{Z}_{1,T}(n)
\end{bmatrix}
= \mathbf{Z}_b(n)
\]

\[
= \mathbf{Z}_b(n)\mathbf{s}(n).
\]

5.3.2. Blind Receiver I/Q Imbalance Compensation. The performance of blind receiver I/Q imbalance compensation algorithm is evaluated next. Here, R&S SMJ acts as transmitter to produce RF signal (with no transmitter IQ imbalance and channel distortions) and MAX2023 to demodulate the signal. The digitized baseband signal (with receiver I/Q imbalance) is transferred to computer through GPIB, where we apply blind DSP algorithm for compensation. The data captured from oscilloscope, however, cannot be used directly for compensation but rather it is first processed to remove the RF chain errors. That involves resampling, delay removal, phase synchronization, offset cancellation, and scaling. After these steps, we apply the compensation algorithm. For illustration purpose, we use a half-loaded OFDM signal with 300 active subcarriers (out of total 1024). The LO frequency is 1.5 GHz. Figure 13 shows the measured spectra before and after compensation, evidencing a clear improvement obtained with the compensation algorithm.

5.3.3. Characterization of Pilot-Based Compensation Schemes. The performance analysis of pilot-based compensation schemes is carried out by creating an OFDM waveform with parameters given in Table 2 as per LTE standard. The
The baseband OFDM signal is sampled at \( 2 \times 1024 \times 15 \text{ KHz} = 30.72 \text{ MHz} \), oversampled by 2, and output by R&S AFQ. The I/Q subsystem of R&S FSG implements the down conversion, filtering, and sampling from 81.6 MHz to \( 2 \times 30.72 \text{ MHz} = 61.44 \text{ MHz} \). Blocks of 512,000 samples are read from R&S FSG memory and loaded into MATLAB memory. The samples are processed for delay estimation, timing synchronization, and LO leakage removal. After the data conditioning, the channel distortions and noise are added, before feeding the signal to R&S SMJ. The CFO effect is also introduced in R&S SMJ by changing the RF frequency. The RF signal is downconverted and sampled at 50 M Samples/second. These samples are extracted from oscilloscope and again processed in MATLAB for resampling to 30.72 MHz, delay estimation, phase synchronization, and scaling. After these processing stages, we have the baseband equivalent model of the signal that has transmitter/receiver I/Q imbalance, channel distortion, noise, and CFO effects included.

The example waveform used for preamble pilot-based compensation algorithms is an OFDM signal with same parameters as were in simulations. In the measurements, we transmit 10 OFDM symbols and use one known OFDM symbol during the estimation and calibration phase. The CFO is 10 KHz and a 3-tap compensation filter for receiver I/Q imbalance and also for transmitter I/Q imbalance estimation. Both ML and ZF equalization schemes are applied to the signal, after receiver I/Q imbalance and CFO compensation. With a fixed channel, the obtained results ensemble averaged over 10 independent measurements are plotted. Figure 14 shows the measurement results of CFO estimation over different SNR values. The results indicate that the estimator performance improves with increasing SNR. Even for small SNR, the MSE remains very low. In Figure 15, measured SER versus SNR is plotted. The figure also shows a reference curve (“No FE Distortion,”) obtained with MATLAB simulation with no I/Q imbalances, perfect CFO, and channel compensation with known estimates. As evident from Figure 15, the algorithms give very consistent performance with SER close to simulated SER. Also, the ML technique provides better performance than ZF technique. Notice that for SNR greater than 26, there is a little performance degradation which is most likely due to measurement noise. In Figure 16, we also plot the symbol constellation after compensation with ZF equalization scheme. The excellent calibration property of algorithms is also visible here.

In the second measurement example, the signal is an OFDM signal having 10 symbols and pilots located at every sixth subcarrier. The signal has altogether 600 active subcarriers out of 1024 subcarriers, with 15 KHz subcarrier spacing. Therefore, in total, there are 1000 pilot symbols. The signal is oversampled by 2. The CFO is now 45 KHz that is greater than the subcarrier spacing. Again the receiver I/Q imbalance is compensated with 3-taps, and the LS and WLS approaches discussed in Section 3 are employed for transmitter I/Q imbalance and channel equalization. The CFO estimation in terms of MSE between the estimated and the true CFO value versus SNR is plotted in Figure 17, which demonstrates a similar behavior to previous preamble pilot-based estimation example. The measured SER versus SNR curve in Figure 18 and symbols after compensation are plotted in Figure 19. The results suggest that the proposed methods are able to deliver very accurate estimates. An important implication of Figure 18 is that with more pilot data available over different symbols, we are able to reduce the effect of measurement noise which was very dominant in the previous measurement example (Figure 15).

In general, it can be concluded from the measurements results of Figures 15 and 18 that the proposed estimator and compensator structures provide considerable improvements to achievable SER, giving confidence to deploy them in practical systems.
5.4. Implementation Challenges. In practice, the measurement setup is exposed to a variety of errors, which must be corrected before introducing the channel distortion and noise as well as before applying the compensation algorithms. We briefly review these errors now.

The downconversion of RF signal (with transmitter I/Q imbalance) using R&S I/Q subsystem is subject to unavoidable residual errors such as synchronization, LO leakage, delay between original and measured signal, nonlinearities of components, and noise of its own. Frequency synchronization between the instruments is achieved by connecting the instruments to the same reference clock of R&S FSG, as shown in Figure 11. Symbol timing synchronization is performed prior to downsampling the signal to symbol rate with ML algorithm. LO leakage results due to finite isolation between the LO and RF ports of the modulator/demodulator. Consequently, when RF signal is up-/downconverted, LO leakage causes offset. It can be mitigated by removing the sample mean of the measured signal. The delay estimation is performed in two steps: first integer delay is removed with
FFT-based correlator, allowing non-data-aided (NDA) ML algorithm for fractional delay estimation and compensation. Detailed review of synchronization algorithms can be found in [36].

Similar to R&S FSG, the data captured from oscilloscope must be DSP conditioned before being used in actual estimation and compensation. The data conditioning involves resampling to original signal sample rate, delay estimation, and offset removal with already-mentioned algorithms.

6. Conclusion

In this paper, the performance of OFDM-based communication systems is studied in the presence of frequency-selective I/Q imbalance, channel distortions and CFO. Based on our signal model, generally applicable algorithms have been developed for the compensation of such impairments. The proposed methods are composed of time domain and frequency domain compensation. It is shown with simulations that the proposed, estimator and compensator structures produce symbol error rates close to the ideal ones. A laboratory measurement setup is also proposed and extensive measurements are carried out to prove the practical value of the algorithms. The measurements results show that the achievable SER is close to the ideal. Hence, the compensation algorithms provide significant improvement to obtainable link performance and can be used in real world radio transceivers. Future work will focus on generalizing the work to MIMO-OFDM and a real-time prototype implementation using FPAGAs for the digital parts and integrated RF-ASICs for the analog circuitry.

Acknowledgments

This work was supported by the Technology Promotion Foundation of Finland (TES), the Nokia Foundation, the graduate school TISE, the Austrian Center of Competence in Mechatronics (ACCM), the Academy of Finland (under the projects “Digitally-Enhanced RF for Cognitive Radio Devices” and “Joint Analysis and DSP-Based Mitigation of Multiple RF Impairments in Future Radio Devices”), and the Finnish Funding Agency for Technology and Innovation (Tekes; under the project “Enabling Methods for Dynamic Spectrum Access and Cognitive Radio”), all of which are gratefully acknowledged. The authors would also like to thank the anonymous reviewers for their helpful comments and suggestions.

References

[1] “Wireless LAN medium access control (MAC) and physical layer (PHY) specifications: high speed physical layer in 2.4 GHz band,” Tech. Rep., IEEE, 1999.
[2] “IEEE standard for local and metropolitan area networks part 16: air interface for fixed broadband wireless access systems,” Tech. Rep., IEEE 802.16, 2004.
[3] U. Ladebusch and C. A. Liss, “Terrestrial DVB (DVB-T): a broadcast technology for stationary portable and mobile use,” Proceedings of the IEEE, vol. 94, no. 1, pp. 183–192, 2006.
[4] G. Faria, J. A. Henriksson, E. Stare, and P. Talmola, “DVB-H: digital broadcast services to handheld devices,” Proceedings of the IEEE, vol. 94, no. 1, pp. 194–209, 2006.
[5] A. B. Ericsson, “Long term evolution (LTE): an introduction,” White Paper, 2007.
[6] S. Mirabbasi and K. Martin, "Classical and modern receiver architectures," IEEE Communications Magazine, vol. 38, no. 11, pp. 132–139, 2000.

[7] A. A. Abidi, "Direct-conversion radio transceivers for digital communications," IEEE Journal of Solid-State Circuits, vol. 30, no. 12, pp. 1399–1410, 1995.

[8] L. Anttila, M. Valkama, and M. Renfors, "Frequency-selective I/Q mismatch calibration of wideband direct-conversion transmitters," IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 55, no. 4, pp. 359–363, 2008.

[9] L. Ding, Z. Ma, D. R. Morgan, M. Zierdt, and G. T. Zhou, “Compensation of frequency-dependent gain/phase imbalance in predistortion linearization systems,” IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 55, no. 1, pp. 390–397, 2008.

[10] M. Valkama, M. Renfors, and V. Koivunen, "Advanced methods for I/Q imbalance compensation in communication receivers," IEEE Transactions on Signal Processing, vol. 49, no. 10, pp. 2335–2344, 2001.

[11] L. Anttila, M. Valkama, and M. Renfors, "Circularity-based I/Q imbalance compensation in wideband direct-conversion receivers," IEEE Transactions on Vehicular Technology, vol. 57, no. 4, pp. 2099–2113, 2008.

[12] L. Anttila, M. Valkama, and M. Renfors, "Efficient mitigation of frequency-selective I/Q imbalance in OFDM receivers," in Proceedings of the IEEE Vehicular Technology Conference (VTC-F '08), Calgary, Canada, September 2008.

[13] L. Anttila and M. Valkama, "On circularity of receiver front-end signals under RF impairments," in Proceedings of the 17th European Wireless Conference, pp. 610–617, Vienna, Austria, April 2011.

[14] L. Anttila, M. Valkama, and M. Renfors, "Blind moment estimation techniques for I/Q imbalance compensation in quadrature receivers," in Proceedings of the IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC '06), Helsinki, Finland, September 2006.

[15] A. Tarighat, R. Bagheri, and A. H. Sayed, "Compensation schemes and performance analysis of IQ imbalances in OFDM receivers," IEEE Transactions on Signal Processing, vol. 53, no. 8, pp. 3257–3268, 2005.

[16] M. Lipardi, D. Mattera, and F. Sterle, "MMSE equalization in presence of transmitter and receiver IQ imbalance," in Proceedings of the International Waveform Diversity and Design Conference (WDD '07), pp. 165–168, Pisa, Italy, June 2007.

[17] A. Tarighat and A. H. Sayed, "Joint compensation of transmitter and receiver impairments in OFDM systems," IEEE Transactions on Wireless Communications, vol. 6, no. 1, pp. 240–247, 2007.

[18] J. Tubbax, A. Fort, L. V. Der Perre et al., "Joint Compensation of IQ Imbalance and Frequency Offset in OFDM systems," in Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '03), vol. 4, pp. 2365–2369, San Francisco, Calif, USA, December 2003.

[19] G. Xing, M. Shen, and H. Liu, "Frequency offset and I/Q imbalance compensation for direct-conversion receivers," IEEE Transactions on Wireless Communications, vol. 4, no. 2, pp. 573–680, 2005.

[20] C. J. Hsu, R. Cheng, and W. H. Sheen, "Joint least squares estimation of frequency, DC offset, I-Q imbalance, and channel in MIMO receivers," IEEE Transactions on Vehicular Technology, vol. 58, no. 5, pp. 2201–2213, 2009.

[21] Y. J. Chiu and S. P. Hung, “Estimation scheme of the receiver IQ imbalance under carrier frequency offset in communication system,” IET Communications, vol. 4, no. 11, pp. 1381–1388, 2010.

[22] G.-T. Gil, I.-H. Sohn, J.-K. Park, and Y. H. Lee, “Joint ML estimation of carrier frequency, channel, I/Q mismatch, and DC offset in communication receivers,” IEEE Transactions on Vehicular Technology, vol. 54, no. 1, pp. 338–349, 2005.

[23] Y.-H. Chung and S.-M. Phoong, "Joint estimation of transmitter and receiver I/Q imbalances, CFO, and channel response for OFDM systems," in Proceedings of the 1st International Conference on Green Circuits and Systems (ICGCS '10), pp. 247–252, Shanghai, China, June 2010.

[24] D. Tandur and M. Moonen, "Joint adaptive compensation of transmitter and receiver IQ imbalance under carrier frequency offset in OFDM-based systems," IEEE Transactions on Signal Processing, vol. 55, no. 11, pp. 5246–5252, 2007.

[25] J. González-Bayón, C. Carreras, and O. Edfors, "A multistandard frequency offset synchronization scheme for 802.11n, 802.16d, LTE, and DVB-T/H systems," Journal of Computer Networks, and Communications, vol. 2010, Article ID 628657, 9 pages, 2010.

[26] T. Roman and V. Koivunen, "Carrier frequency synchronization for mobile television receivers," in Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS '07), pp. 261–264, May 2007.

[27] A. Kayani, L. Anttila, Y. Zou, and M. Valkama, "Hybrid time/frequency domain compensator for RF impairments in OFDM systems," in Proceedings of the 22nd IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC '11), Toronto, Canada, September 2011.

[28] M. Speth, S. Fechtel, G. Fock, and H. Meyr, “Optimum receiver design for OFDM-based broadband transmission—part II: a case study,” IEEE Transactions on Communications, vol. 49, no. 4, pp. 571–578, 2001.

[29] J. Younghwan, K. Jihyun, L. Yuro, A. Jaemin, C. Wongyu, and L. Dongchan, "Additional diversity gain in OFDM receivers under the influence of IQ imbalances," in Proceedings of the IEEE International Conference on Communications (ICC '07), pp. 5915–5920, Glasgow, Scotland, June 2007.

[30] S. Narayanan, B. Narasimhan, and N. Al-Dhahir, "Digital baseband compensation of joint TX/RX I/Q imbalance in mobile MIMO SC-FDE transceivers in the presence of CFO," in Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '10), pp. 3510–3513, March 2010.

[31] J. G. Proakis, Digital Communications, McGraw-Hill, New York, NY, USA, 3rd edition, 1995.

[32] L. Deneire, P. Vandenameele, L. Van der Perre, B. Gyselinckx, and M. Engels, "A low-complexity ML channel estimator for OFDM," IEEE Transactions on Communications, vol. 51, no. 2, pp. 135–140, 2003.

[33] S. Haykin, Adaptive Filter Theory, Prentice-Hall, Upper Saddle River, NJ, USA, 4th edition, 2002.

[34] T. B. Sorensen, P. E. Mogensen, and F. Frederksen, "Extension of the ITU channel models for wideband (OFDM) systems," in Proceedings of the IEEE Vehicular Technology Conference (VTC '05), pp. 392–396, Dallas, Tex, USA, September 2005.

[35] 3GPP Technical Specification Group Radio Access Network, "Evolved universal terrestrial radio access (E-UTRA) and evolved universal terrestrial radio access (E-UTRAN); overall description; stage 2," Tech. Rep. TS 36.300, V1.0.0, 2007.

[36] U. Mengali and A. N. D. Andrea, Synchronization Techniques for Digital Receivers, Plenum Press, New York, NY, USA, 1997.
1. Introduction

To increase the transmission capacity, multiple-input multiple-output (MIMO) systems have been proposed for the next generation wireless communication systems, and therefore the need for a high-performance and low-complexity MIMO detector becomes an important issue. The maximum likelihood (ML) detector is known to be an optimal detector; however, it is impractical for realization owing to its great computational complexity. Addressing this problem, researchers have proposed tree-based search algorithms, such as sphere decoding [1] and K-Best decoding [2], to reduce the complexity with near-optimal performance. On the other hand, channel matrix preprocessing technique, such as lattice-reduction-aided (LRA) detection [3], has been proposed to improve the MIMO detection performance.

The lattice reduction transforms the channel matrix into a more orthogonal one by finding a better basis for the same lattice so as to improve the diversity gain of the MIMO detector. The Lenstra-Lenstra-Lovász (LLL) algorithm is a well-known lattice reduction algorithm for its polynomial execution time. In the literature [4], the LLL algorithm is widely employed to improve the lattice-reduction MIMO detection or to reduce the MIMO detection complexity. However, the LLL algorithm has many redundant check operations that have never been addressed in the literature. These redundant operations lead to many unnecessary computations and thus increase the processing latency and complexity. Therefore, we propose a look-ahead check technique to detect and avoid the unnecessary check operations in the LLL algorithm. This technique not only generates the lattice-reduced matrix which obeys the size reduction and LLL reduction in the original LLL algorithm but also applies to real- and complex-value LLL algorithm [5].

The remainder of this paper is organized as follows. Section 2 briefly describes the signal model for MIMO detection. In Section 3, we introduce the lattice-reduction-aided MIMO detection and the LLL algorithm. In Section 4, we demonstrate the proposed LR-LLL algorithm, and in Section 5 we present the simulation and analysis results. The corresponding hardware architecture and processing cycle counts estimation is shown in Section 6. Finally, we summarize our conclusions in Section 7.
where $x \in \mathbb{A}^N$ is the transmitted signal vector, $y \in \mathbb{C}^N$ is the received signal vector, $H = [h_1, h_2, \ldots, h_{Ny}]$ represents a flat-fading channel matrix, and $n \in \mathbb{C}^N$ is the white Gaussian noise with variance $\sigma_n^2$. All the vectors $h_i$ are independent and identically distributed complex Gaussian random vectors with zero means and unit variances. Set $\mathbb{A}$ consists of the constellation points of the QAM modulation. Then, we re-formulate the equivalent real channel matrix as follows:

\[
y_r = \begin{bmatrix}
\Re(y)
\mathcal{J}(y)
\end{bmatrix} = Hx_r + n_r
\]

\[
= \begin{bmatrix}
\Re(H)
-\mathcal{J}(H)
\end{bmatrix}
\begin{bmatrix}
\Re(x)
\mathcal{J}(x)
\end{bmatrix} + \begin{bmatrix}
\Re(n)
\mathcal{J}(n)
\end{bmatrix}.
\]

The QR decomposition is often applied in the pre-processing of the MIMO detection because it provides decoding efficiency. Then, the channel matrix $H_r$ can be expressed by

\[
H_r = Q_rR_r,
\]

where $Q_r \in \mathbb{R}^{N \times M}$ is an orthogonal matrix and $R_r \in \mathbb{R}^{M \times M}$ is an upper triangular matrix. By multiplying $Q_r^T$ on both sides of (2), we can obtain

\[
\hat{y}_r = Q_r^Ty_r = R_r x_r + Q_r^Tn_r,
\]

where $Q_r^Tn_r$ is white Gaussian. In addition, we adopt column-norm-based sorted QR decomposition (SQRD) [6] because it not only enhances detection performance but also reduces the computational complexity of the lattice reduction [7].

### 3. Lattice Reduction

A lattice $\mathbb{L}$ is defined as \{\(t_1h_1 + t_2h_2 + \cdots + t_Nh_N\mid t_1 \cdots t_N \in \mathbb{Z}\), where \(\{h_1, \ldots, h_N\} \in \mathbb{R}^N\)\} are the basis vectors. The lattice reduction algorithm aims to find a unimodular matrix $T (\det T = 1$ and all elements of $T$ are integers) such that a more orthogonal $H_r = HT$ has the same lattice as $H_r$. Then, the signal model becomes

\[
y = Hx_r + n = \hat{H}_rT^{-1}x_r + n_r = \hat{H}_r s + n_r.
\]

If $x_r \in \mathbb{Z}^N$, $T^{-1}x_r = s \in \mathbb{Z}^N$. In practice, the transmitted signals $x_r$ do not belong to an integer set; however, we can still transform the signals $x_r \in \mathbb{A}^N$ into an integer set by linear operations such as scaling and shifting.

Several lattice-reduction algorithms are described in the literature, and the LLL algorithm [11] is the most popular approach. Because QR preprocessing is often employed in the MIMO detector, the LLL algorithm is then modified for Q and R matrices [12], as shown in Algorithm 1. In the literature, lines (4) to (19) are often defined as a loop that can be decomposed into two parts: (1) lines (4) to (10) deals with the size reduction operations; and (2) lines (11) to (19) handle LLL reduction operations. The number of iterations performed in the size reduction depends on the index $k$, and the LLL reduction operation may increase or decrease the index $k$ depending on the result of the LLL reduction check ($\delta(\|R(k - 1, k - 1)\|) > |\Re(k, k)|^2 + |\Re(k - 1, k)|^2$). Therefore, the number of loops certainly depends on the values in the $R$ matrix, and thus the processing latency varies for different channel matrices. Moreover, we find that most of the computational complexity is contributed by the operations when the check conditions ($\mu \neq 0$ for size reduction and $\delta(\|R(k - 1, k - 1)\|) > |\Re(k, k)|^2 + |\Re(k - 1, k)|^2$ for LLL reduction check) are satisfied; that is, the size and
MIMO detection [7]. In order to eliminate the redundant computation complexity and latency in the literature on LRA, we propose a look-ahead check technique by classifying the original loops to check operations in the LLL algorithm, we propose a look-ahead check technique. We calculate the percentage of the redundant decrease of $k$ achieves 67% for $2 \times 2$ lattice reduction and converge to 28% because the size and LLL reductions have been performed. Thus, the decrease of $k$ is not always necessary because the size and LLL reductions have been checked in the last loop. We can see that the percentage of the redundant decrease of $k$ achieves 67% for $2 \times 2$ lattice reduction and converge to 28% if the MIMO dimension is larger than $10 \times 10$. Therefore, we propose a look-ahead check technique to modify index $k$ and avoid unnecessary check operations in the original LLL algorithm.

### 4. Look-Ahead Check

The number of loops is often treated as a benchmark for computation complexity and latency in the literature on LRA MIMO detection [7]. In order to eliminate the redundant check operations in the LLL algorithm, we propose a look-ahead check technique by classifying the original loops to forward loop and back loop. And the loop reduction LLL algorithm is shown in Algorithm 2. The corresponding flow chart of the proposed algorithm and each loop is shown in Figure 1.

#### 4.1. Back Loop

We define the back loop as the loop that only contains LLL reduction check and LLL violation processing as shown in Figure 1. We find that the size reduction constraint will not be violated after the $k$ is decreased because the $R(k,k)$ has already been size reduced in the previous processing and is not changed in the column-swapping operation. And the givens rotation will only change the $k-1$ row while the row value above $k-1$ row remains size reduced. That means only LLL reduction check is required in the back loop. This LLL reduction check is named as back-state LLL reduction check to differentiate with the origin one. Because only the LLL reduction part needs to be executed in this back loop, we use a while loop in our algorithm to avoid the redundant size reduction operation. Nonetheless, there is still one case that the size reduction will process. If the division result exactly equals 0.5, the original LLL algorithm will do the size reduction operation in the back loop. But our algorithm will skip. This will produce a different lattice-reduced matrix at last. However, to do the size reduction or not to do in this case will both produce the matrix that obeys the LLL lattice reduction criterion. Although, we cannot prove the performance is the same mathematically, we will show that their performance is the same through the Monte-Carlo simulation in latter section. So the lattice reduction would differentiate with the original one.

| Antenna number ($N \times N$) | $N = 2$ | $N = 4$ | $N = 6$ | $N = 8$ | $N = 10$ | $N = 12$ | $N = 14$ | $N = 16$ |
|-------------------------------|--------|--------|--------|--------|--------|--------|--------|--------|
| Percentage of no operation   | 67%    | 40%    | 33%    | 30%    | 29%    | 28%    | 28%    | 28%    |

**Algorithm 2:** The proposed loop reduction LLL algorithm using look-ahead check technique.

- **INPUT:** $Q, R$, such that $H = QR$
- **OUTPUT:** $Q, R, T$ such that $HT = QR$

1. Initialize $Q = R$, $R = R$, $T = I, a$
2. $k = 2$
3. While $k \leq N$
4. For $p = k - 1, \ldots, 1$
5. $\mu = \|R(p, k)/R(p, p)\|$ If $\mu \neq 0$
6. $R(1 : p, k) \leftarrow R(1 : p, k) - \mu R(1 : p, p)$
7. $T(:, k) \leftarrow T(:, k) - \mu T(:, p)$
8. End
9. End
10. If $\delta|R(k - 1, k - 1)|^2 > |R(k, k)|^2 + |R(k - 1, k)|^2$
11. Do
12. Swap columns $k - 1$ and $k$ in $R$ and $T$
13. Calculate Givens rotation matrix $\Theta = \left( \begin{array}{cc} a & b \\ -b & a \end{array} \right)$
14. $a = \|R(k - 1, k, k - 1)\|$ and $b = \|R(k - 1, k, k - 1)\|$
15. $R(1 : k - 1 ; : ) \leftarrow R(1 : k - 1 ; : ) \Theta^T$
16. $Q(:, k - 1 : k) \leftarrow Q(:, k - 1 : k) \Theta^T$
17. $k \leftarrow k - 1$
18. While $\delta|R(k - 1, k - 1)|^2 > |R(k, k)|^2 + |R(k - 1, k)|^2$ & $k \neq 1$
19. If $|R(k, k + 1)/R(k, k)| \neq 0$
20. $k \leftarrow k + 1$
21. Else
22. $k \leftarrow k + 2$
23. End
24. Else
25. $k \leftarrow k + 1$
26. End
27. End

**Table 1:** The percentage of no operation after index $k$ is decreased.
performance will not suffer any degradation. Using the look-ahead check technique, we can more precisely determine the next \( k \) value at the end of each loop.

4.2. Forward Loop. Forward loop is just like the original loop defined in the previous section except once the LLL reduction is violated, it will enter the back loop. If back-state LLL reduction constraint is not violated, we enter the stay-state size reduction check to predict the next index \( k \). Notice that if the stay-state size reduction constraint \( (\mu \neq 0) \) is not violated, the LLL reduction constraint must not be violated because the \( R(k,k) \) has already been LLL reduced in the previous processing and all values remain unchanged. Therefore, we can also perform the stay-state size reduction check ahead. If the stay-state size-reduction constraint is not violated, we can simply increase the index \( k \) by 2 to skip a redundant LLL reduction and enter the forward loop.

5. Simulation Results

To verify the proposed LLL algorithm, we simulate the LLL-aided MIMO detections based on the MIMO system described in Section 2, and we employ sorted QR decomposition in all MIMO detectors. The LLL-reduction parameter \( \delta \) equals 0.75, as suggested in [11]. Table 2 shows the average loop numbers of the original and the proposed LLL algorithms for different antenna numbers. Forward loop and back loop are all counted as a loop in our algorithm. The proposed LLL algorithm can reduce the average number of loops to 93% ~ 94% of the original LLL algorithm. The BER versus SNR curve is shown for 4 × 4 and 8 × 8 MIMO systems in Figures 2 and 3, respectively. The performance is exactly the same for our algorithm and the original LLL algorithm.

We also analyze the computational complexity and latency of our algorithm. The results for 4 × 4 and 8 × 8 MIMO systems are listed in Tables 3 and 4. The computation is divided to four operations such as addition, multiplication, division, and givens rotation. Our algorithm is lower in total computational complexity and especially in the division which tends to cost more time for computation. The lower ratio is just like the loop-reduced ration. But only average computational complexity cannot clearly show the advantage of our algorithm. Since the original LLL algorithm contains lots of redundant checks operation which are unable to process in parallel, it will result in long average processing time to complete the lattice reduction operation. We try to simulate the latency by parallelizing all the possible operations. The latency counts are as follows: the line (5) to line (9) in our algorithm is counted as one division, one addition and one multiplication. The LLL reduction check operation contains four multiplications and two additions. The column swap operation is counted as no operation delay. The givens rotation counts one at each back loop. And the stay-state size...
Table 3: Average computational complexity and latency of lattice-reduction Algorithms for 4 × 4 MIMO detection.

| Algorithm | Addition | Multiplication | Division | Givens Rotation | Total |
|-----------|----------|----------------|----------|----------------|-------|
| LLL       | 204.6/49.4 | 238.5/83.4 | 64.45/64.45 | 67.34/10.43 | 575/208.4 |
| LR-LLL    | 202.79/47.5 | 234.81/79.5 | 46.43 (72%)/46.43 | 67.83/10.43 | 551.88 (96%)/184 (88.2%) |

Table 4: Average computational complexity and latency of lattice-reduction Algorithms for 8 × 8 MIMO detection.

| Algorithm | Addition | Multiplication | Division | Givens Rotation | Total |
|-----------|----------|----------------|----------|----------------|-------|
| LLL       | 1463.3/141.6 | 1541.8/220 | 339.6/339.6 | 285.8/24.34 | 3630/725.6 |
| LR-LLL    | 1462.5/137.21 | 1527.4/211.1 | 218.12 (64%)/218.12 | 287.46/24.49 | 3504.5 (96%)/591 (81.4%) |

6. Hardware Architecture

6.1. Top Structure. In this paper, we proposed a very intuitive structure for our LR-LLL algorithm in Figure 4. The center controller counts the index k by the LLL violation results. And it will send control signals to choose the specific matrix element to the input of the combinational circuit. We can also call size reduction part as size reduction loop and LLL reduction part as LLL reduction loop, respectively. The update circuit for the remaining R matrix, T matrix, and Q matrix are omitted for simplification. In this architecture, CORDIC circuit has two pipelined stages. So it required one cycle for size reduction loop and four cycles for a LLL reduction loop. The traditional LLL algorithm always processes the forward loop which contains the execution of the whole circuit. While using LR-LLL algorithm, some forward loops will replace by back loops. The average cycle counts for the LLL algorithm, and our LR-LLL algorithm is listed in Table 5. We can find out that as the antenna number grows, the reduction of average cycle grows. And the FPGA results are shown in Table 6. In [5], the complex-valued LLL lattice reduction algorithm is proved to have lower computational complexity than real-valued system. This is mainly due to the double sized of the real number system comparing to complex number. So the hardware or cycle counts may be larger than the previous two complex number works.

Table 5: Average Cycle of Lattice Reduction Algorithms for different antenna numbers.

| N  | Original LLL algorithm | LR-LLL algorithm |
|----|------------------------|------------------|
| 4  | 109                    | 89 (81.6%)       |
| 8  | 451                    | 320 (70.9%)      |
| 12 | 869                    | 469 (53.9%)      |

6.2. Other Blocks. The divr block executes the divide and round operation which can be easily designed by long...
Figure 4: Hardware architecture for LR-LLL algorithm.

Table 6: FPGA implementation results and comparison.

|                | Virtex-4 [8] | Vertex-4 [9] | Virtex-4 [10] | Virtex-4 [This work] |
|----------------|--------------|--------------|---------------|---------------------|
| LRA algorithm  | RS-LLL       | CLLL (modified) | CT-LLL       | LR-LLL             |
| System model   | Complex      | Complex      | Real          | Real               |
| Slices         | 4805         | 3640         | 11330         | 4170               |
| Antenna number | 4            | 4            | 4             | 4                  |
| Frequency      | 79 MHz       | 140 MHz      | 8.7 MHz       | 53 MHz             |
| Ave. cycles/matrix | 14          | 96           | 5             | 89                 |
| Throughput (M mat/s) | 5.6        | 1.45         | 1.74          | 0.59               |

Figure 5: Division and round circuit.

division architecture. In Figure 5, we show a four-stage long-division architecture for five bits output divr circuit. The size reduction update circuit is composed of multiplication and addition circuit. Instead of calculating the square norm to do the LLL reduction comparison, we choose the CORDIC vector mode circuit to calculate the square root of the norm which may also be the output if the LLL check violates. The square root of $\delta$ is set to 0.875 to approximate the square root of 0.7. CORDIC rotation mode is used to do the Givens rotation of the algorithm. The output of the comparison circuit is the LLL reduction violation check results which will control the center controller and also enable the update circuits. The LLL reduction update block contains multiple CORDIC rotation circuits to do the givens rotation of remaining row element of $R$ and $Q$ matrix. It also contains a swap circuit for $T$ matrix.

7. Conclusion

In this paper, we propose a look-ahead check technique to eliminate unnecessary check operation in the LLL algorithm. The proposed algorithm not only reduces the average number of loops in the LLL algorithm but also reduces the computation complexity and latency of LLL algorithm. We also proposed a very intuitive architecture to estimate the clock cycle saving of our algorithm. The saving is dramatically increased while the antenna number grows. Therefore, we believe that the proposed loop reduction LLL algorithm benefits the lattice-reduction-aided MIMO detection.

References

[1] E. Agrell, T. Eriksson, A. Vardy, and K. Zeger, “Closest point search in lattices,” *IEEE Transactions on Information Theory*, vol. 48, no. 8, pp. 2201–2214, 2002.
[2] M. Shabany and P. G. Gulak, “The application of lattice-reduction to the K-Best algorithm for near-optimal MIMO detection,” in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS ’08)*, pp. 316–319, May 2008.
[3] H. Yao and G. Wornell, “Lattice-reduction-aided detectors for MIMO communication systems,” in *Proceedings of the IEEE Global Telecommunications Conference*, vol. 1, pp. 424–428, November 2002.

[4] W. Zhang, X. Ma, B. Gestner, and D. V. Anderson, “Designing low-complexity equalizers for wireless systems,” *IEEE Communications Magazine*, vol. 47, no. 1, pp. 56–62, 2009.

[5] Y. H. Gan, C. Ling, and W. H. Mow, “Complex lattice reduction algorithm for low-complexity full-diversity MIMO detection,” *IEEE Transactions on Signal Processing*, vol. 57, no. 7, pp. 2701–2710, 2009.

[6] P. Luethi, A. Burg, S. Haene, D. Perels, N. Felber, and W. Fichtner, “VLSI implementation of a high-speed iterative sorted MMSE QR decomposition,” in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS ’07)*, pp. 1421–1424, May 2007.

[7] M. Sandell, A. Lillie, D. McNamara, V. Ponnampalam, and D. Milford, “Complexity study of lattice reduction for MIMO detection,” in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC ’07)*, pp. 1088–1092, March 2007.

[8] L. Bruderer, C. Studer, M. Wenk, D. Seethaler, and A. Burg, “VLSI implementation of a low-complexity LLL lattice reduction algorithm for MIMO detection,” in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS ’10)*, pp. 3745–3748, May 2010.

[9] B. Gestner, Z. Wei, X. Ma, and D. V. Anderson, “Lattice reduction for MIMO detection: from theoretical analysis to hardware realization,” *IEEE Transactions on Circuits and Systems—Part I: Regular Papers*, vol. 58, no. 4, pp. 813–826, 2011.

[10] C. F. Liao and Y. H. Huang, “Power-saving 4x4 lattice-reduction processor for MIMO detection with redundancy checking,” *IEEE Transactions on Circuits and Systems—Part II: Express Briefs*, vol. 58, no. 2, pp. 95–99, 2011.

[11] A. K. Lenstra, H. W. Lenstra, and L. Lovasz, “Factoring polynomials with rational coefficients,” *Mathematische Annalen*, vol. 261, no. 4, pp. 515–534, 1982.

[12] D. Wübben, R. Bohnke, V. Kuhn, and K.-D. Kammeyer, “MMSE-based lattice-reduction for near-ML detection of MIMO systems,” in *Proceedings of the ITG Workshop on Smart Antennas*, pp. 106–113, May 2004.
Research Article

Probabilistic Behavior Analysis of MIMO Fading Channels under Geometric Mean Decomposition

Ping-Heng Kuo and Pang-An Ting

Information and Communications Research Laboratories, Industrial Technology Research Institute (ITRI), Chutung, Hsinchu 31040, Taiwan

Correspondence should be addressed to Ping-Heng Kuo, pinghengkuo@itri.org.tw

Received 13 June 2011; Accepted 22 September 2011

Academic Editor: Yin-Tsung Hwang

Copyright © 2012 P.-H. Kuo and P.-A. Ting. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Geometric mean decomposition (GMD) has been proposed as a method to realize multiple spatial links with identical gains that are intrinsic to a MIMO channel. In order to simplify system design and implementation based on knowledge regarding probability behavior of MIMO-GMD schemes, the main objective of this paper is to statistically characterize the link gains and channel capacities that can be provided via GMD. In particular, closed-form univariate and bivariate probability density functions (PDFs) for these metrics under Rayleigh fading are derived using Gamma approximations. By applying these analytical results, the fluctuations of MIMO-GMD schemes are examined by modeling both link gains and capacities using finite-state Markov chains (FSMCs).

1. Introduction

Numerous research results have shown that spatial multiplexing-based multiple-input multiple-output (MIMO) transceivers can boost the data rate of wireless communication systems without using additional power and channel bandwidth. When both transmitter and receiver are equipped with multiple antennas, multiple independent data streams can be concurrently launched into the air by the transmitter and separately decoded at the receiver, with aid of certain signal processing techniques [1]. By combining MIMO with the multicarrier schemes such as orthogonal frequency division multiplexing (the so-called MIMO-OFDM), channel capacity can be tremendously increased as the degree of freedom in frequency and spatial domain are jointly utilized.

By assuming that channel state information (CSI) is available at both ends of the link, singular value decomposition (SVD) [2] is commonly considered as the approach to extract spatial links within a MIMO channel. With SVD, the MIMO channel matrix can be transformed into a bank of parallel scalar air-pipes (widely termed as eigenmodes in the literature) by applying appropriate beam-forming matrices at both transmitter and receiver. Since the eigenmodes realized by SVD are equivalent to singular values of the MIMO channel matrix, their link strengths could be quite different. Therefore, proper bit-allocation and power distribution (such as water-filling) algorithms must be employed to optimize the system capacity. This, however, increases the complexity in adaptive control mechanism, as bit/power allocations among eigenmodes may have to be updated frequently in rapidly time-varying channels. For the sake of convenience, one may simply apply the modulation/coding format or distribute the power uniformly on all eigenmodes. Nevertheless, this may cause degradation on error performance of the system, especially when magnitude of the minimum channel singular value is too small [3].

Geometric mean decomposition (GMD), proposed by Jiang et al. [4], has emerged as an alternative method of designing MIMO transceiver. A set of parallel spatial links with identical gains intrinsic to a MIMO channel can be realized via GMD. Remarkably, these parallel links with the identical gain are equal to the geometric mean of channel channel singular values (eigenmodes). The property of parallel links with identical gain makes GMD useful for adaptive MIMO-OFDM systems, as complicated joint bit/power allocation...
algorithms in frequency and spatial domain can be significantly simplified. In recent years, much attention from both academia and industry has been paid to the GMD-based MIMO transceiver. For instance, GMD has been deemed as a prospective MIMO architecture in some next-generation cellular broadband networks, including two of the most important standards in this context—LTE-Advanced [5] and WiMAX (IEEE 802.16 m) [6]. Note that Jiang et al. have put forward another MIMO transceiver dubbed as uniform channel decomposition (UCD) [7], which also provides spatial links with identical gains. The scope of this paper is focussed on GMD, and the related work on UCD is, however, remained as open problems for future research.

In order to calculate performance metrics such as bit error rate and outage probability of a GMD-based MIMO transceiver operating in mobile fading channels, analytical results on statistical properties of MIMO eigenmode geometric mean are required. To the best of our knowledge, however, only a handful of papers in the existing literature have offered the relevant analysis. In [8], asymptotic behavior of geometric mean of MIMO channel eigenvalues has been examined. Also, the probability density function (PDF) for the link gain of MIMO-GMD in special cases with two antennas at both transmitter and receiver can be found in [9], but the results for more general cases are not available. This motivated us to derive closed-form expressions of PDFs (either exact or approximated) for systems with arbitrary number of antennas in Rayleigh fading channels. Specifically, apart from the univariate PDFs, we are also interested in the bivariate PDFs for MIMO eigenmode geometric mean and channel capacity, since the statistical behavior relating to channel variation is crucial for systems incorporating with adaptive transmission techniques. For instance, knowledge on how channel quality evolves with time allows engineers to determine the adaptation/feedback rate for MIMO-GMD in a more judicious manner. Further, for successive interference cancelation (SIC) detection algorithm in conventional MIMO-GMD transceivers [4], the reliability of a data stream is dependent on previously decoded data streams. That is, if the receiver fails to detect a data stream correctly, the data on the remaining streams may not be decoded successfully due to error propagation. Thus, it is possible to gauge how frequently that an error propagation event would occur if time-varying characteristics are known. In correspondence, this paper applies first-order finite-state Markov chains (FSMC) [10, 11] to model the fluctuations of both link gain and channel capacity under MIMO-GMD in baseline cases of Rayleigh fading environments. Note that FSMC has been used to emulate the capacity process of general MIMO wireless channels [12], but channels with MIMO-GMD transceiver have not been considered. Also, some researchers have designed the feedback mechanisms for adaptive modulation systems with channels modeled by FSMC [13]. Based on the univariate and bivariate PDFs developed in this paper, the transition probabilities among the discrete states of an FSMC are analytically computed.

In a nutshell, this paper encompasses the following contributions.

(1) Derivation of the exact PDFs for link gains and channel capacities of MIMO-GMD systems with two antennas at either transmitter or receiver and an arbitrary number of antennas on the other side.

(2) Demonstration of Gamma approximations for link gains of MIMO-GMD systems with arbitrary numbers of antenna at either transmitter or receiver.

(3) Derivation of PDFs for channel capacities of MIMO-GMD systems with arbitrary numbers of antenna at either transmitter or receiver, based on Gamma approximation of link gains.

(4) Extending the Gamma approximation to provide bivariate PDFs for link gains and channel capacities of MIMO-GMD systems with numbers of antenna at either transmitter or receiver.

(5) Constructing first-order FSMCs for both link gains and channel capacities of MIMO-GMD systems. The transition probabilities among multiple quantized Markov states are computed based on the derived PDFs.

The rest of the paper is organized as follows. In Section 2, assumptions on channel model and background of the MIMO-GMD architecture are briefly reviewed. Then, in Section 3, the closed-form expressions of both univariate and bivariate PDFs for eigenmode geometric mean and channel capacity of MIMO-GMD schemes are derived. Based on the derived analytical results, FSMCs are constructed in Section 4 to study the time variation of MIMO channel under GMD. Finally, this paper ends with a concise conclusion drawn in Section 5.

2. Background and Assumptions

In this section, we first describe the considered system model. Then, the concept of GMD and the fundamental structure of an FSMC are reviewed subsequently.

2.1. Channel Model. In general, this paper mainly studies a \((N_t,N_r)\) single-user, point-to-point MIMO system that has \(N_t\) transmit antennas and \(N_r\) receive antennas. The MIMO channel, \(H\), is therefore an \(N_r \times N_t\) matrix. Here, we denote \(m = \min(N_t,N_r)\) and \(n = \max(N_t,N_r)\). We presume this MIMO system is coupled with OFDM setting, so channel response per subcarrier is basically flat in frequency domain. Therefore, the signal model for each subcarrier can be written as:

\[
y = Hx + z, \tag{1}
\]

where \(y\) is an \(N_r \times 1\) vector of the received signal, \(x\) is an \(N_t \times 1\) vector of the transmitted data symbols, and \(z\) is an \(N_r \times 1\) vector of additive complex zero mean Gaussian noise. Since Rayleigh fading is considered, all entries of \(H\) are modeled as complex Gaussian random variables with zero mean and unit variance, \(\mathcal{CN}(0,1)\). With SVD, the MIMO channel \(H\) can be written as:

\[
H = U \Sigma V^\dagger, \tag{2}
\]
where both \( U \) and \( V \) are unitary matrices, \((\cdot)^T\) denotes Hermitian transpose, and \( \Sigma \) is a diagonal matrix with \( m \) non-zero elements representing the singular values of \( H : s_1, s_2, \ldots, s_m \). The unordered joint density for the singular values of \( H \) is [14]

\[
f(s_1, \ldots, s_m) = \frac{2^m}{m! \prod_{i=1}^{m} (n-i)! (m-i)!} \times \exp\left(-\sum_{i=1}^{m} s_i^2\right) \times \prod_{i=1}^{m} s_i^{2(n-1)} \prod_{i<j} (s_i^2 - s_j^2)^2,
\]

where \( l = n - m \). Note that the eigenvalues of the channel correlation matrix, \( \lambda_i = s_i^2 \), are equivalent to power gains of the spatial links intrinsic to the MIMO channel, and their joint statistics are governed by the central Wishart distribution as given in [15]

\[
f(\lambda_1, \ldots, \lambda_m) = \frac{1}{m! \prod_{i=1}^{m} (n-i)! (m-i)!} \times \exp\left(-\sum_{i=1}^{m} \lambda_i\right) \times \prod_{i=1}^{m} \lambda_i^{(n-1)} \prod_{i<j} (\lambda_i - \lambda_j)^2.
\]

It is further assumed that the MIMO channel evolves over time in accordance to

\[
H(t+\tau) = J_0(2\pi f_D \tau)H(t) + \sqrt{1 - J_0(2\pi f_D \tau)^2} \Psi,
\]

where \( J_0(\cdot) \) is the zeroth-order Bessel function, \( f_D \) and \( \tau \) are the Doppler frequency and time displacement, respectively, \( \tau \) is a \( N_r \times N_t \) random matrix with zero-mean, unit variance complex Gaussian elements. For the sake of simplicity, we denote the autocorrelation function as \( \xi(\tau) = J_0(2\pi f_D \tau) \) in the remainder of this paper.

### 2.2. Review of Geometric Mean Decomposition

The authors of [4] suggested designing MIMO transceivers via geometric mean decomposition (GMD), in which the MIMO channel is decomposed as:

\[
H = QRP^T,
\]

where \( P \) and \( Q \) are semiunitary matrices and \( R \) represents an \( m \times m \) upper triangular matrix with identical diagonal elements. If the data signal \( x \) is precoded by \( P \) at the transmitter and \( y \) is filtered by \( Q^T \) at the receiver, the signaling model (1) is tantamount to multiplication between \( x \) and the upper triangular matrix, \( R \). Hence, the transmitted signal could be extracted and decoded using nulling and cancellation procedure as elucidated in [4]. Assuming ideal operations without error propagation, the data signals are sent on \( m \) parallel spatial links with a common value of amplitude:

\[
g_{\text{SR}} = \left( \prod_{l=1}^{m} s_l \right)^{1/m}.
\]

In other words, the amplitudes of these links are equivalent to the geometric mean of MIMO channel singular values, \( s_i \). Since these \( m \) spatial links have an identical gain, the “worst subchannel” problem, in which the overall error performance is dominated by the weakest spatial link, is no longer a concern. Thus, one merit of using GMD is the exemption of complicated bit/power allocations over spatial links. By uniformly allocating the transmission power among all parallel spatial links realized by GMD, the channel capacity can be expressed as:

\[
C = m \log_2(1 + y g_{\text{SR}}^2) = m \log_2(1 + y g^2),
\]

where \( y \) denotes the signal-to-noise ratio (SNR) on each of the parallel links. Also, it is easy to see that the link power gain of these spatial links is the geometric mean of \( \lambda_s \):

\[
g = g_{\text{SR}}^2 = \left( \prod_{l=1}^{m} \lambda_l \right)^{1/m}.
\]

Conventionally, the channel capacity of a general MIMO system can be written as the sum of multiple random variables; hence, central-limit theorem (CLT) could be used to approximate the MIMO channel capacity as a Gaussian random variable [16]. For MIMO-GMD schemes, however, Gaussian approximation may not be suitable as (8) is simply a scalar integer multiple of a single random variable.

### 2.3. Finite-State Markov Chain

As one of the objectives in this paper is to model the time-varying behavior of MIMO-GMD systems based on FSMC, the fundamental structure of an FSMC is outlined here for pedagogical reasons. To model the fluctuation of a random process using an FSMC, the process is initially quantized into a number of discrete states. Then, the dynamic behavior is described by the transition probabilities among these states. For simplicity, only first-order FSMC is considered in this paper, which means the transition probability is solely dependent on the most recent observation. In our case, both processes of eigenmode geometric mean and capacity are quantized based on their magnitudes. In particular, the random process of interest is simply partitioned into \( S \) states by setting \( S - 1 \) threshold levels. In the first state (denoted as \( S_0 \)), the process has a value smaller than the lowest threshold level \( T_0 \). In the final state (denoted as \( S_{n-1} \)), the process is larger than the highest threshold level \( T_{S-2} \). Otherwise, the process is said to be in the \( i \)th state (denoted as \( S_i \)) if its value falls in the range between threshold levels \( T_{i-1} \) and \( T_i \). For an illustrative example, Figure 1 shows an eigenmode geometric mean process that is partitioned into four states using three threshold levels: \( T_0, T_1, \) and \( T_2 \). As shown in the later sections of this paper, the transition probabilities can be computed based on both univariate and bivariate PDFs of the process.

### 3. Derivations of Statistical Distributions

This section targets to derive univariate and bivariate PDFs for both spatial link power gain, \( g \), and channel capacity, \( C \),
for MIMO-GMD schemes. For univariate distributions, the exact analytical results are first presented for cases with two spatial links \( (m = 2) \), and then the cases with more spatial links are dealt with by Gamma approximations.

3.1. Univariate Distributions for Cases with \( m = 2 \). When either the transmitter or receiver is equipped with only two antennas, only \( m = 2 \) spatial links are realizable as the rank of a MIMO channel is \( \min(N_t, N_r) \). From (3), the joint PDF for MIMO channel singular values with \( m = 2 \) is

\[
f(s_1, s_2) = \frac{2 \exp(-s_1^2 - s_2^2)}{(n-1)!(n-2)!} s_1^{n-3} s_2^{n-3} (s_1^2 - s_2^2)^2.
\]

Since the geometric mean, \( g_s \), is a function of singular values, Jacobian transform can be applied to obtain the joint PDF for \( s_2 \) and \( g_s \). We have

\[
s_1 = \frac{g_s^2}{s_2} \quad \Rightarrow \quad \frac{\partial s_1}{\partial g_s} = \frac{2g_s}{s_2},
\]

and hence

\[
f(s_2, g_s) = \left| \frac{\partial s_1}{\partial g_s} \right| f(s_1, s_2)|_{s_1 = s_1(g_s)} = 4g_s \exp\left[\frac{-(g_s^2/s_2)^2}{s_2} \right] \frac{(s_1^2 - s_2^2)^2}{s_2(1)(n-2)!}.
\]

The PDF for \( g_s \) can be acquired by integrating (12) with respect to \( s_2 \). After tedious calculation and little algebra, we find the following result:

\[
f(g_s) = \int_0^\infty f(s_2, g_s) \, ds_2 = \frac{8g_s^{3n-3}}{(n-1)!(n-2)!} K_1(2g_s^2).
\]

where \( K_1(\cdot) \) represents modified Bessel function of the second kind of order 1. The derivation from (12) to (13) could be carried out by certain symbolic manipulation software packages. Note that (13) is a generalized version for cases with \( m = 2 \); by setting \( n = 2 \), (13) can be reduced to the special case result for a \( (2,2) \) MIMO-GMD system given in [9]. As \( g_s \) is the square root of \( g \), we may apply simple transformation on (13) to get the PDF for \( g \):

\[
f(g) = \frac{4g^{2(n-1)}}{(n-1)!(n-2)!} K_1(2g).
\]

Similarly, Jacobian transform can be employed to cope with the channel capacity of MIMO-GMD schemes, as the PDF for \( g \) is now readily available. Based on (8), we have

\[
g = \frac{2^{C/m} - 1}{y} \Rightarrow \frac{\partial g}{\partial C} = \frac{\ln(2)2^{C/m}}{my}.
\]

Thus, the PDF for channel capacity with \( m = 2 \) is

\[
f(C) = \left| \frac{\partial g}{\partial C} \right| f(g)|_{g\to g(C)} = 2\ln(2)\sqrt{\frac{C}{2}} - 1 \frac{2^{n-1}}{y^{n-0.75}(n-1)!(n-2)!} K_1\left(\frac{2}{y}(2^{C/2} - 1)\right).
\]

The validity of (14) and (16) can be verified through Monte Carlo simulations. In Figure 2, the empirical distributions of eigenvalue geometric mean for \( (2,2) \), \( (2,4) \), and \( (2,8) \) MIMO systems are compared with their respective analytical PDFs that we have derived (14). On the other hand, the comparison between theoretical channel capacity PDFs (16) and simulation results are shown in Figure 3. In all cases, excellent agreements between analytical and simulation results can be observed.

3.2. Univariate Distributions for Cases with \( m > 2 \). For systems with more than two spatial links, that is, \( m > 2 \), the similar approach invoked in Section 3.1 could be used. Nonetheless, it is difficult to get closed-form results like (14) and (16) because multiple integrals are required in the step of obtaining \( f(g_s) \) from \( f(s_2, \ldots, s_m, g_s) \) as in (13). Fortunately, some previous studies have concluded that MIMO eigenvalues \( (\lambda_1, \lambda_2, \ldots, \lambda_m) \) can be very accurately approximated by Gamma random variables [14]. Moreover, [17] claims that the geometric mean of multiple independent Gamma random variables could be either a Gamma or a mixture of Gamma distributions. Although MIMO eigenvalues are not independent processes, they are weakly correlated. Thus, we may make a hypothesis stating that

\[
f(g) \approx \Gamma(a, \theta) \\
\]

where the parameters \( a = E(g)/\theta \) and \( \theta = \text{var}(g)/E(g) \) are shape factor and scale factor of Gamma distribution respectively, and \( \Gamma(\cdot) \) represents Gamma function. Since \( \text{var}(g) = E(g^2) - E(g)^2 \), the computation for \( \text{var}(g) \) requires the first
two moments of $g$. Note that the $t$-th moment of $g$ can be calculated as

$$E(g^t) = \int_0^\infty \cdots \int_0^\infty \left( \prod_{i=1}^m \lambda_i \right)^{t/m} f(\lambda_1, \ldots, \lambda_m) d\lambda_1, \ldots, d\lambda_m,$$

where $f(\lambda_1, \ldots, \lambda_m)$ is the joint PDF of all $m$ eigenvalues of the channel correlation matrix $H^\dagger H$, as given in (4).

To justify the suitability of Gamma approximations for eigenvalue geometric means, similar comparisons as in Section 3.1 are carried out for (4,4), (4,8), and (8,8) cases in Table 1. Hence, one may construct the PDFs for $g$ in these MIMO cases accordingly.

The approximations based on Gamma distributions (17) can be further leveraged to find the PDFs for channel capacity in MIMO-GMD schemes. Using (15), we have

$$f(C) = \frac{\partial g}{\partial g} \times f(g) \big|_{g = g(C)},$$

where $g$ is given in (19). To make this paper more self-contained, we have tabulated the numerical values of $E(g), \text{var}(g), k, \text{and } \theta$ for (4,4), (4,8), and (8,8) cases in Table 1. Hence, one may construct the PDFs for $g$ in these MIMO cases accordingly.

| MIMO Configuration | $E(g)$ | $\text{var}(g)$ | $k$ | $\theta$ |
|--------------------|--------|----------------|----|----------|
| (4,4)              | 1.8075 | 6.0279         | 3.2150 | 0.5507 |
| (4,8)              | 6.1880 | 23.7393        | 18.7687 | 0.1713 |
| (8,8)              | 2.921  | 0.2539         | 0.1713 | 0.2921 |

Table 1: Parameters for Gamma approximation for some selected MIMO configurations.

3.3. Bivariate Distributions. Since we have observed that the MIMO-GMD link gains can be accurately approximated by a Gamma random variable, it can be conjectured that a bivariate Gamma PDF is well suited for the joint density of their magnitudes at two correlated time instants, $g(t)$ and $g(t + \tau)$. Thus, by modifying the bivariate Gamma PDF given in [18], we have

$$f(g, \tilde{g}) \approx \frac{(\theta g)^{(k-1)/2}}{\theta^{k+1} \Gamma(k)(1-\rho)^{(k-1)/2}} \exp \left( \frac{-g + \tilde{g}}{\theta(1-\rho)} \right) \times I_{k-1} \left( \frac{2}{\theta(1-\rho)} \sqrt{\rho g \tilde{g}} \right),$$

where $g$ and $\tilde{g}$ are the joint PDFs of their magnitudes at two correlated time instants, and $I_k(\cdot)$ denotes the modified Bessel function of the first kind of order $k$.
In (21), both \(E(g)\) and \(\text{var}(g)\) can be calculated by utilizing (4) and (18). Since \(g\) and \(\hat{g}\) are functions of \((\lambda_1(t), \ldots, \lambda_m(t))\) and \((\lambda_1(t + \tau), \ldots, \lambda_m(t + \tau))\), respectively, the computation for the joint moment, \(E(\hat{g}g)\), requires the joint density for MIMO eigenvalues at two time instants, which has been provided in [19]:

\[
f(\lambda_1, \ldots, \lambda_m, \hat{\lambda}_1, \ldots, \hat{\lambda}_m) = \prod_{i<j} \left( \frac{1}{\alpha^2} \left( \hat{\lambda}_i - \hat{\lambda}_j \right) \right) \prod_{i} \left( \lambda_i - \lambda_i \right) \prod_{i=1}^{m} \lambda_i^{\alpha} \prod_{i=1}^{m} \left( n - i \right)! \left( m - i \right)! \exp \left( -\sum_{i=1}^{m} \lambda_i \right) \times G(\lambda),
\]

where

\[
G(\lambda) = \det \begin{bmatrix}
    f(\hat{\lambda}_1 | \lambda_1) & f(\hat{\lambda}_1 | \lambda_2) & \cdots & f(\hat{\lambda}_1 | \lambda_m) \\
    f(\hat{\lambda}_2 | \lambda_1) & f(\hat{\lambda}_2 | \lambda_2) & \cdots & f(\hat{\lambda}_2 | \lambda_m) \\
    \vdots & \vdots & \ddots & \vdots \\
    f(\hat{\lambda}_m | \lambda_1) & \cdots & \cdots & f(\hat{\lambda}_m | \lambda_m)
\end{bmatrix},
\]

\[
f(\hat{\lambda} | \lambda) = \frac{\alpha^2}{\beta^2} \left( \frac{\hat{\lambda}}{\alpha^2 \lambda} \right)^{v/2} \exp \left[ -\frac{\alpha^2 \lambda - \hat{\lambda}}{\beta^2} \right] I_l \left( 2 \frac{\alpha}{\beta^2} \sqrt{\lambda \hat{\lambda}} \right)
\]

with \(\hat{\lambda}_t = \lambda_i(t + \tau), \alpha = \xi, \beta = \sqrt{1 - \xi^2}\), and \(\xi = \frac{f_0}{2\pi f_D \tau}\) as before. The joint moment, \(E(\hat{g}g)\), can thereby be computed with (22) as

\[
E(\hat{g}g) = \int_{0}^{\infty} \cdots \int_{0}^{\infty} \int_{0}^{\infty} \left( \prod_{i=1}^{m} \lambda_i \right)^{1/m} \left( \prod_{i=1}^{m} \hat{\lambda}_i \right)^{1/m} \\
\times f(\lambda_1, \ldots, \lambda_m, \hat{\lambda}_1, \ldots, \hat{\lambda}_m) \, d\lambda_1, \ldots, d\lambda_m, \hat{\lambda}_1, \ldots, \hat{\lambda}_m.
\]

Then, \(\rho\) can be calculated by substituting (24) into (21). Apparently, the closed-form expression for (24) is difficult, if not impossible, to obtain. However, it still can be solved via numerical integrations.

Once the value of \(\rho\) is acquired, (20) can be extended to obtain the bivariate PDF for MIMO-GMD channel capacity.
process. Using Jacobian transform, \( f(C, \hat{C}) \) can be written as
\[
f(C, \hat{C}) = \det \begin{bmatrix} \frac{\partial g}{\partial C} & \frac{\partial \hat{g}}{\partial C} \\ \frac{\partial g}{\partial \hat{C}} & \frac{\partial \hat{g}}{\partial \hat{C}} \end{bmatrix} \times f(g, \hat{g}) \bigg|_{g=g(C), \hat{g}=\hat{g}(C)}
\]
\[
= 0.4805 \times 2^{(C+\hat{C})/m} \left( \frac{2^{C/m} - 1}{2^{\hat{C}/m} - 1} \right)^{(k-1)/2} \times \frac{m^2 \varpi^2 \Gamma(k) \theta^{k+1}(1 + \rho)}{2^{2(C/m - 1)/(2^{C/m} - 1)}}
\]
\[
\times \exp \left[ \frac{2 - 2^{C/m} - 2^{\hat{C}/m}}{\varpi^2(1 - \rho)} \right] \times w_{k-1} \left[ \frac{2\sqrt{\rho}\left(2^{C/m} - 1\right)}{\varpi^2(1 - \rho)} \right].
\]

With PDFs derived in this section, one may construct an FSMC to model the fluctuation of MIMO-GMD link gain and capacity, as explained in Section 4.

4. FSMC Construction

In order to investigate the time-varying behavior of MIMO-GMD systems, FSMC is adopted to model both link gain and channel capacity. Based on the FSMC structure elaborated in Section 2.3, we delineate how transition probabilities can be analytically calculated using the PDFs derived in Section 3. The transition probabilities from \( \delta_i \) to \( \delta_j \), denoted as \( P_{i,j} \), for the link gain process \( g(t) \) can be computed as
\[
P_{i,j} = \frac{\text{Prob}(g(t) \in \delta_i, g(t+\tau) \in \delta_j)}{\text{Prob}(g(t) \in \delta_i)}, \tag{26}
\]
where
\[
\text{Prob}(g(t) \in \delta_i, g(t+\tau) \in \delta_j) = \int_{\delta_i} \int_{\delta_j} f(g, \hat{g}) dg \, d\hat{g}, \tag{27}
\]
\[
\text{Prob}(g(t) \in \delta_i) = \int_{\delta_i} f(g) dg. \tag{28}
\]

Note that \( f(g) \) has been derived as (14) or (17), depending on the value of \( m \). Analytical calculations of (27) can be quite difficult, so numerical integrations are used in this work instead. For the sake of simplicity, this paper assumes that the channel variation is slow enough so the process would only transit to one of the adjacent states (from \( \delta_i \) to \( \delta_{i-1} \) or \( \delta_{i+1} \)) or stay in the same state (from \( \delta_i \) to \( \delta_i \)). Therefore, we have
\[
P_{i,i} = 1 - P_{i,i+1} - P_{i,i-1}. \tag{29}
\]

For channel capacity in MIMO-GMD scheme, the transition probabilities can be computed in a similar fashion.
in this work. In practical adaptive modulation schemes, for example, the threshold levels for state quantization could be set based on minimum required channel gain for a target error performance. In all simulations, we have \( f_D = 30 \text{ Hz} \), \( \tau = 0.001 \text{ sec} \) and \( y = 10 \text{ dB} \). In Figure 6, the link gain of a \((2,4)\) MIMO-GMD scheme is modeled as an FSMC consisting of four states with \( \{T_0, T_1, T_2\} = \{1.5, 2.5, 3.5\} \), and transition probabilities from both simulation and calculations are plotted. Similarly, we approximate the MIMO-GMD capacity process in a \((2,2)\) system using a four-state FSMC with \( \{T_0, T_1, T_2\} = \{4, 7, 9\} \) bps/Hz in Figure 7. In both cases, it is apparent that our calculations can provide very accurate approximations.

5. Conclusion

By applying GMD to a MIMO-OFDM system, multiple spatial links with identical gains can be realized within each subcarrier. This assuages the complexity at the transmitter side as the spatial domain bit/power allocation can be simplified. In order to seek for potential opportunities to further decrease the system complexity, this paper has investigated the statistical properties of MIMO systems using GMD. In particular, we have derived PDFs for link gains (MIMO eigenvalue geometric mean) and capacities. Although exact results are only available for cases with two parallel links, Gamma approximations can be used to model eigenvalue geometric means for general MIMO cases with arbitrary antenna configurations. Moreover, these results are extended to derive the bivariate PDFs, which is important for the analysis of time-varying behavior. These results are employed to model the channel variation in MIMO-GMD schemes by constructing FSMCs. To be specific, the transition probabilities among states could be computed using the PDFs derived in this paper. This paper proposed a potential approach to predict the fluctuation of MIMO channels under GMD, which may allow the engineers to relent the feedback rate and hence reduce the system burden. The analytical PDF results presented in this paper are for baseline cases of Rayleigh flat fading. For scenarios involve spatial correlation and Ricean fading, statistical properties of MIMO-GMD schemes are still open problems and should be addressed in the future.

References

[1] P. W. Wolniansky, G. J. Foschini, G. D. Golden, and R. A. Valenzuela, “V-BLAST: an architecture for realizing very high data rates over the rich-scattering wireless channel,” in Proceedings of the IEEE International Symposium on Signals, Systems and Electronics (ISSSE ’98), pp. 295–300, Pisa, Italy, September 1998.

[2] G. Lebrun, J. Gao, and M. Faulkner, “MIMO transmission over a time-varying channel using SVD,” IEEE Transactions on Wireless Communications, vol. 4, no. 2, pp. 757–764, 2005.

[3] G. Burel, “Statistical analysis of the smallest singular value in MIMO transmission systems,” in Proceedings of the International Conference on Signal, Speech, and Image Processing (WSEAS ’02), Skiathos, Greece, September 2002.

[4] Y. Jiang, J. Li, and W. W. Hager, “Joint transceiver design for MIMO communications using geometric mean decompositions,” IEEE Transactions on Signal Processing, vol. 53, no. 10, pp. 3791–3803, 2005.

[5] “GMD beamforming for LTE-advanced SU/MU-MIMO,” in R1-093799, 3GPP TSG RAN WG1 Meeting 58, Miyazaki, Japan, October 2009.

[6] “A joint transceiver design for MIMO precoder and antenna selection mechanism,” in C802.16m-08/314r2, IEEE Broadband Access Working Group TGm Session 55, Macau, China, May 2008.

[7] Y. Jiang, J. Li, and W. W. Hager, “Uniform channel decomposition for MIMO communications,” IEEE Transactions on Signal Processing, vol. 53, no. 11, pp. 4283–4294, 2005.

[8] J. Salo, D. Seethaler, and A. Skupch, “On the asymptotic geometric mean of MIMO channel eigenvalues,” in Proceedings of the IEEE International Symposium on Information Theory (ISIT ’06), pp. 2109–2113, Seattle, WA, USA, July 2006.

[9] D. Tang and M.-X. Lv, “Analysis of multiuser diversity in geometric mean decomposition-based MIMO system,” in Proceedings of the International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM ’07), pp. 61–64, Shanghai, China, 2007.

[10] H. S. Wang and N. Moayeri, “Finite-state Markov channel—a useful model for radio communication channels,” IEEE Transactions on Vehicular Technology, vol. 44, no. 1, pp. 163–171, 1995.

[11] Q. Zhang and S. A. Kassam, “Finite-state markov model for rayleigh fading channels,” IEEE Transactions on Communications, vol. 47, no. 11, pp. 1688–1692, 1999.

[12] S. Vaihunthan, S. Haykin, and M. Sellathurai, “MIMO channel capacity modeling using Markov models,” in Proceedings of the 61st IEEE Vehicular Technology Conference (VTC ’05-Spring), pp. 126–130, Spring, Stockholm, Sweden, April 2005.

[13] A. E. Ekpenyong and Y. F. Huang, “Feedback-detection strategies for adaptive modulation systems,” IEEE Transactions on Communications, vol. 54, no. 10, pp. 1735–1740, 2006.

[14] P.-H. Kuo, Fluctuations of MIMO Wireless Channels: Analysis based on Brownian Motion Theory, Dr. Muller, Saarbrucken, Germany, 2008.

[15] E. Telatar, “Capacity of multi-antenna Gaussian channels,” European Transactions on Telecommunications, vol. 10, no. 6, pp. 585–595, 1999.

[16] P. Smith and M. Shafi, “An approximate capacity distribution for MIMO systems,” IEEE Transactions on Communications, vol. 52, no. 6, pp. 887–890, 2004.

[17] R. S. Laurent, “The distribution of the geometric mean of independent F (or gamma) random variables,” in Proceedings of the Joint Statistical Meetings (JSM ’01), Atlanta, GA, USA, August 2001.

[18] G. E. Öien, H. Holm, and K. J. Hole, “Impact of channel prediction on adaptive coded modulation performance in Rayleigh fading,” IEEE Transactions on Vehicular Technology, vol. 53, no. 3, pp. 758–769, 2004.

[19] P.-H. Kuo, P. J. Smith, and L. M. Garth, “Joint density for eigenvalues of two correlated complex wishart matrices: characterization of MIMO systems,” IEEE Transactions on Wireless Communications, vol. 6, no. 11, pp. 3902–3906, 2007.