This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2014.2312727, IEEE Transactions on Vehicular Technology

A Robust Decoding Scheme for Convolutionally Coded Transmission Through a Markov Gaussian Channel

Fikreselam Gared Mengistu, Der-Feng Tseng, *Member*, *IEEE*, Yunghsiang S. Han, *Fellow*, *IEEE*, Megistu Abera Mulatu, and Li-Chung Chang, *Member*, *IEEE*

***Abstract*—Communication systems are susceptible to impulse noise, particularly when the impulse statistics are not time-invariant and are difficult to accurately model. To address the challenge of impulse noise, a robust and efficient decoding scheme was devised for single carrier convolutionally coded transmissions over memory impulse noise channels. By accommodating channel states, but without relying on statistical knowledge of impulses, the Viterbi algorithm (VA) based on an expanded set of trellis states, was employed to perform maximum likelihood decoding. A detailed analysis of complexity was offered; the analytical results reinforced the efficiency of the proposed scheme compared with the traditional VA. The simulation results indicated that the proposed decoding scheme is compellingly robust: the bit error probability performance level attained using the proposed decoder is remarkably close to that of an optimal decoder, which uses impulse statistics; furthermore, the proposed decoder was superior to the alpha-penalty function decoder, which neglects the channel memory property and experiences an error floor, in fairly general circumstances.**

***Index Terms*—Impulse noise, Markov Gaussian channel, transition probability, Viterbi algorithm (VA).**

I. Introduction

IMPULSE noise, typically non-Gaussian, is commonly encountered in both wireless [1], [2] and wired communication systems [3]; for instance, electromagnetic interference originating from the electronic equipment in power line communications (PLC) is typically characterized by bursts and extremely strong instantaneous power relative to the additive white Gaussian noise (AWGN). The negative effects of non-Gaussian noises on system performance have received considerable interest in the communication community, prompting investigation into communication system designs and their performance levels when using non-Gaussian noise models.

When designing communication systems, impulse noises are typically modeled using either a memoryless channel (memoryless channels cannot represent bursts of interferences), or a memory channel. The memoryless channel models of impulse noise have been widely studied in the Class-A [4]–[6] and Bernoulli Gaussian models [7] in PLC. The impulse noise, modeled using a uniform distribution, was addressed in [8] by using energy detection to conduct spectrum sensing in a cognitive radio (CR) system. [9] proposed a novel receiver, using a factor graph approach for orthogonal frequency division multiplexing (OFDM) transmission in an impulse noise environment. Robust clipping for OFDM transmissions over memoryless impulse noise channels was investigated in [10], without using the a priori knowledge of the impulse noise probability density function (PDF). [11] analyzed the performance of space-time coding in OFDM systems subjected to fading and impulse noise. Furthermore, decoders have been extensively evaluated using various efficient decoding metrics in impulse noise environments [12]–[17]. Misusing a memoryless channel model in the decoder causes inferior performance levels compared with decoders that exploit the inherent memory channel. The bursts inherent to this impulse noise channel can be modeled using a two-state Markov Gaussian process [18].

Researchers have studied coded communications systems through memory channels based on various assumptions at the decoder. The research performed has been summarized as follows: based on iterative receivers, the achievable information rates of the considered communication systems were measured in [19]; numerous decoding metrics involving the Viterbi algorithm (VA) through a Markov Gaussian channel were applied and compared in [20]; a robust unknown interference detection method for use in OFDM-based CR was devised in [21]; and [22] evaluated the performance of a convolutionally coded narrowband system through a memory channel that is modeled by a partitioned Markov chain. Because impulse statistics, such as channel states and the PDF of impulses, are unlikely to be known at the receiver, various decoding metrics for convolutionally coded transmissions over a Markov Gaussian channel were examined in [20] without using the impulse noise statistics. In this study, a new, efficient, and robust decoding metric for single carrier communication systems is proposed that does not rely on the impulse noise statistics or channel state information in the memory channel. The proposed efficient decoding method involves using a maximum likelihood codeword search that can be conducted similarly to the VA; furthermore, to consider the channel states of impulse noise, virtual states between two encoder states were included in the trellis diagram. Calculations of the computational complexity are provided, elucidating the feasibility of the proposed decoding algorithm. A performance

The authors are with the Department of Electrical Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan. (e-mail: dtseng@mail.ntust.edu.tw.)

This work was supported by the National Science Council of Taiwan under NSC-101-2221-E-011-072-MY2, NSC 101-2221-E-011-069-MY3, and NSC 101-2221-E-011-073.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2014.2312727, IEEE Transactions on Vehicular Technology

comparison was conducted using computer simulations of the proposed scheme, a maximum likelihood decoder, and a decoder that employed the alpha-penalty function decoder (alpha-PFD) metric [20].

To clarify, an example of the CR system in which the proposed scheme can likely be used is elucidated as follows. When unlicensed users access licensed bands, the degree to which the signals of primary users deteriorate should be assessed to ensure that the radio frequency spectrum [23] is efficiently used; CR systems that exclusively focuses on suppressing background noise without continually addressing the interference from other current users are prone to performance degradation [24], [25]. Spectrum sensing, which is optional based on the regulations of the Federal Communications Commission, can address this problem at the expense of receiver complexity, and employing a mandatory database that provides information, such as the periodicity of the primary user transmissions, may alleviate the performance loss to a certain extent; thus, secondary users of the CR system attempt to occupy the frequency bands at low power profiles [26]. It is plausible that shadowing effects and a lack of sophistication could cause incongruous scheduling by the CR coordinator, leading to the coexistence of signals from both licensed and unlicensed users. From this perspective, the strong signals of primary users critically distort those of the secondary users because licensed users likely become suddenly active on the same frequency bands. This negative effect of interference on signal recovery is a critical and challenging factor [27]. Furthermore, the behavior of this interference matches the characteristics of impulse noise [28]: short in duration, it exhibits ON/OFF-like activity and its statistics, which are characterized by a power spectral density (PSD) that typically varies over time because of randomness in the system configuration pose extreme difficulties when attempting to construct accurate model. Thus, the proposed decoder was developed to suppress intruding interference without relying on assumed statistics, providing an efficient tool to compensate for the performance loss in unideal CR systems.

The remainder of this paper is organized as follows. System and noise models are described in Section II. The details of the proposed decoding algorithm, including the decoder metric (and its primary comparative counterpart) and its computational complexity, are presented in Section III. The simulated results are presented in Section IV, and a conclusion is offered in Section V.

II. System Model

A convolutionally coded communication system is briefly introduced, followed by a description of a memory impulse noise model that employs the widely adopted Markov Gaussian channel. Fig. 1 shows a block diagram of the system model.

A. Convolutionally Coded Communication Systems

A (n_{cc}, k_{cc}, m) convolutional code (CC) exhibiting an information sequence of length $k_{cc}L$ bits was used, where n_{cc} code bits are produced blockwise in response to each block input of k_{cc} information bits, and m is the memory length of the convolutional encoder. L is the number of $k_{cc}-$ bit blocks in the information sequences. Because m blocks of k_{cc}-bit zeros are appended to the end of an information sequence to clear the contents of shift registers, the length of each codeword is $N = n_{cc}(L + m)$. Thus, an interleaver was used in system model, where the size of the block interleaver is reliant on the maximal acceptable transmission delay.

Assuming that the modulation format is binary phase-shift keying (BPSK),[1] the received symbol sequence $\boldsymbol{y} = (y_0, y_1, \ldots, y_{N-1})$ can be represented as follows:

$$\boldsymbol{y} = (-1)^{\boldsymbol{v}}\sqrt{E_s} + \boldsymbol{n}, \tag{1}$$

where $\boldsymbol{v} = (v_0, v_1, \ldots, v_{N-1}) \in \{0, 1\}^N$ is the transmitted codeword and E_s is the energy of the modulated symbol. The noise term $\boldsymbol{n}$, which is characterized by impulse occurrence (in addition to background noise) is detailed in the following section.

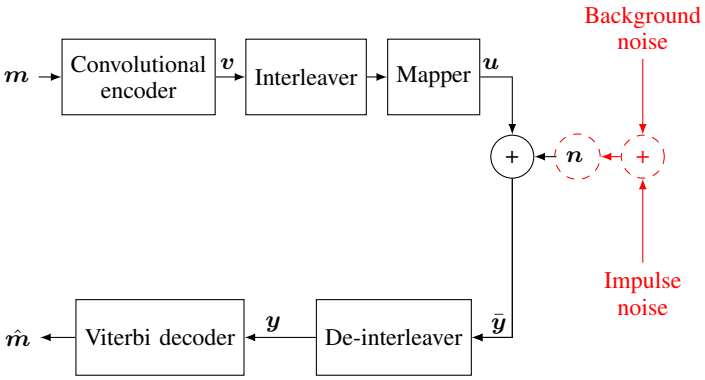


Fig. 1. System model of single carrier convolutionally coded transmissions over a Markov Gaussian channel

B. Noise Model

The statistical properties of each noise sample n_j are defined by the channel state $s_j \in \{G, B\}$, which can be categorized as either a good (G) or a bad (B) channel state. If $s_j \in G$, the noise term contains only AWGN; however, if $s_j \in B$, the noise is the sum of the impulsive interference and the AWGN. Typically AWGN noise $\{g_j\}_{j=0}^{N-1}$ exhibits a flat single-sided PSD of height N_0. The PDF distributions of bad and good channel states differ according to their variances, considering that the Gaussian random variable of the noise is conditioned based on the channel state [7]; thus, the Gaussian distribution noise PDF of real-valued transmissions for the (G) and (B) channel states are expressed as follows:

$$\begin{aligned} \Pr(n_j|s_j = G) &= \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-n_j^2}{2\sigma^2}\right) \\ \Pr(n_j|s_j = B) &= \frac{1}{\sqrt{2\pi R\sigma^2}} \exp\left(\frac{-n_j^2}{2R\sigma^2}\right), \end{aligned} \tag{2}$$

[1]The derivation is directly applicable to quadrature phase-shift keying (QPSK) with gray mapping, of which in-phase and quadrature components are equivalent to two BPSK symbols. In addition, the derivation can be similarly obtained for higher-order M-ary modulation schemes.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2014.2312727, IEEE Transactions on Vehicular Technology

where σ^2 and R are the variance in background Gaussian noise when the channel state is good, and the average noise power ratio between the bad and good channel states, respectively. $R = 1 + \frac{\sigma_I^2}{\sigma^2}$ is the parameter that quantifies the strength of impulse noise and is likely to be large, where σ_I^2 is the variance in impulse noise. To accommodate the memory property inherent to the strong impulsive interference, the noise state sequence is modeled based on a first-order two-state Markov process [19]; furthermore, the transformation of noise sample n_j in a noise PDF is driven by four transition probabilities, which can be calculated based on the bad and good channel states, as shown in Fig. 2.

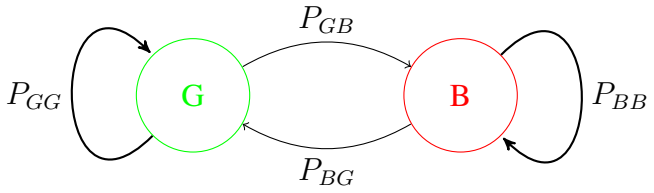


Fig. 2. Two-state Markov model of channel state transitions

Assuming a stationary and irreducible Markov chain, the transition probability matrix is constructed as follows:

$$Q = \begin{bmatrix} P_{GG} & P_{GB} \\ P_{BG} & P_{BB} \end{bmatrix} = \begin{bmatrix} P_G & P_B \\ P_G & P_B \end{bmatrix} + \eta \begin{bmatrix} P_B & -P_B \\ -P_G & P_G \end{bmatrix}, \tag{3}$$

where $P_{s_j s_{j+1}} = \Pr(s_{j+1}|s_j)$ is the transition probability from channel state $s_j \in \{G, B\}$ to $s_{j+1} \in \{G, B\}$, $P_{s_j} = \Pr(s_j)$ is the channel state probability, and $\eta = 1 - \frac{P_{BG}}{P_G} = 1 - \frac{1}{\gamma}$. The parameter $\gamma = \frac{P_G}{P_{BG}}$ is used to measure the channel memory [18]; when $\gamma = 1$ ($\eta = 0$), all transition probabilities are replaced by state probabilities ($P_{GG} = P_{BG} = P_G$ and $P_{BB} = P_{GB} = P_B$), and the channel becomes memoryless. By contrast, $\gamma \neq 1$ is used to quantify memory, which can be categorized as persistent memory ($\gamma > 1$ or $\eta > 0$) or oscillatory memory ($\gamma < 1$ or $\eta < 0$) [19]. Furthermore, the average burst length of impulse B_L is related to P_{BG}; $B_L = \frac{1}{P_{BG}}$. Because this study was conducted to address bursty impulse noises, only persistent memory ($\gamma > 1$ or $\eta > 0$) was considered when developing the decoding algorithm. An interleaver, preceded by a convolutional encoder, was used to disperse the bursty impulse noise, dampening its negative effects on performance. The interleaver depth I can yield the following transition matrix [20]:

$$Q^I = \begin{bmatrix} P_G & P_B \\ P_G & P_B \end{bmatrix} + \eta^I \begin{bmatrix} P_B & -P_B \\ -P_G & P_G \end{bmatrix}, \tag{4}$$

where $\eta^I = 1 - I\frac{P_{BG}}{P_G}$.

Before implementing the proposed decoding algorithm, it must be emphasized that impulse statistics are unavailable to the receiver. Therefore, the notations that include a superscripted "(d)", such as $P_B^{(d)}(P_G^{(d)})$, $P_{GG}^{(d)}$, $P_{GB}^{(d)}$, $P_{BG}^{(d)}$, $P_{BB}^{(d)}$, and $R^{(d)}$, are assumed channel parameters at the decoder, and may differ from the actual channel parameters.

III. Proposed Robust Decoding Metric

This section describes the derivation of the proposed metric and an expanded trellis, which is proposed to account for the channel states of noise, such that maximum likelihood decoding is conducted similarly as the VA. The computational complexity of the proposed decoding algorithm is also described.

A. Proposed Robust Decoding Metric

This study involved designing and efficiently implementing a robust metric for use in a convolutional decoder through the Markov Gaussian channel. Before proceeding with the derivation, notations must be defined: $\boldsymbol{y}_0^j$, $\boldsymbol{v}_0^j$, and $\boldsymbol{s}_0^j$ are the received samples, codeword bits, and channel states regarding time from 0 to j, respectively. When $j = N-1$, $\boldsymbol{y}_0^{N-1}$, $\boldsymbol{v}_0^{N-1}$, and $\boldsymbol{s}_0^{N-1}$ are simply denoted using $\boldsymbol{y}$, $\boldsymbol{v}$, and $\boldsymbol{s}$, respectively. The set of all codewords is denoted as $\boldsymbol{C} = \{\boldsymbol{v}_0, \ldots, \boldsymbol{v}_{2^{k_{cc}L}-1}\}$. Note that $\boldsymbol{y}$ is dependent on $\boldsymbol{v}$ and $\boldsymbol{s}$. The expression $(\hat{\boldsymbol{v}}, \hat{\boldsymbol{s}}) \in \boldsymbol{C} \times \{G, B\}^N$ denotes the joint maximum-likelihood decision over all possible pairs $(\tilde{\boldsymbol{v}}, \tilde{\boldsymbol{s}}) \in \boldsymbol{E}_N$, where $\boldsymbol{E}_N$ is the collection of all $(\tilde{\boldsymbol{v}}, \tilde{\boldsymbol{s}})$ pairs, and codeword $\tilde{\boldsymbol{v}}$ is paired with channel state $\tilde{\boldsymbol{s}} = (\tilde{s}_0, \tilde{s}_1 \ldots, \tilde{s}_{N-1})$. The term v_j denotes the code bit that was actually transmitted, $\tilde{v}_j$ denotes the code bit associated with the branch of the trellis currently decoded, and $\hat{v}_j$ is the decoding decision. The same conventions apply to the notations for channel states.

The maximum a posteriori probability (MAP) decoding rule suggests that decision $(\hat{\boldsymbol{v}}, \hat{\boldsymbol{s}}) \in \boldsymbol{E}_N$ is made if

$$\Pr(\boldsymbol{y}|\hat{\boldsymbol{v}}, \hat{\boldsymbol{s}}) \Pr(\hat{\boldsymbol{v}}, \hat{\boldsymbol{s}}) \geq \Pr(\boldsymbol{y}|\tilde{\boldsymbol{v}}, \tilde{\boldsymbol{s}}) \Pr(\tilde{\boldsymbol{v}}, \tilde{\boldsymbol{s}}) \tag{5}$$

holds for all $(\tilde{\boldsymbol{v}}, \tilde{\boldsymbol{s}}) \in \boldsymbol{E}_N$. Because the transmitted codeword and channel state are independent, $\Pr(\tilde{\boldsymbol{v}}, \tilde{\boldsymbol{s}}) = \Pr(\tilde{\boldsymbol{v}}) \Pr(\tilde{\boldsymbol{s}})$. Furthermore, assuming equal a priori probabilities among the information sequences, $\Pr(\tilde{\boldsymbol{v}}) = \frac{1}{2^{k_{cc}L}}$ for all $\tilde{\boldsymbol{v}} \in \boldsymbol{C}$. Based on (5):[2]

$$\begin{aligned} &\sum_{j=0}^{N-1} \ln\left[\Pr(y_j|\hat{v}_j, \hat{s}_j) \Pr(\hat{s}_j|\hat{s}_{j-1})\right] \\ \geq &\sum_{j=0}^{N-1} \ln\left[\Pr(y_j|\tilde{v}_j, \tilde{s}_j) \Pr(\tilde{s}_j|\tilde{s}_{j-1})\right], \end{aligned} \tag{6}$$

where $\tilde{s}_{-1}$ is the initial channel state. Thus, the right side of (6) acts as the path metric of $(\tilde{v}_j, \tilde{s}_j)$ for the proposed decoding metric, and the term inside the summation is the bit metric of $(\tilde{v}_j, \tilde{s}_j)$ which is expressed as follows:

$$\beta(y_j|\tilde{v}_j, \tilde{s}_j, \tilde{s}_{j-1}) = \ln[\Pr(y_j|\tilde{v}_j, \tilde{s}_j)] + \ln[\Pr(\tilde{s}_j|\tilde{s}_{j-1})]. \tag{7}$$

The decoding rule considers all possible transition probabilities for the corresponding channel states; with the noise statistics available at the optimal decoder (the PDF of the recieved sample conditioned on the codeword bit for both the good and bad channel states are calculated in (28) and (29),

[2]The detailed derivation from (5) to obtain (8) is given in Appendix.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2014.2312727, IEEE Transactions on Vehicular Technology

respectively), the decoding rule becomes the following :

$$\begin{aligned}
&\sum_{j=0}^{N-1}(1-b_j)\left\{\ln\left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)-\frac{(y_j-(-1)^{\hat{v}_j}\sqrt{E_s})^2}{2\sigma^2}\right.\\
&\qquad+(1-b_{j-1})\ln(\Pr(\hat{s}_j=G|\hat{s}_{j-1}=G))\\
&\qquad\left.+b_{j-1}\ln(\Pr(\hat{s}_j=G|\hat{s}_{j-1}=B))\right\}\\
&+\sum_{j=0}^{N-1}b_j\left\{\ln\left(\frac{1}{\sqrt{2\pi R\sigma^2}}\right)-\frac{(y_j-(-1)^{\hat{v}_j}\sqrt{E_s})^2}{2R\sigma^2}\right.\\
&\qquad+(1-b_{j-1})\ln(\Pr(\hat{s}_j=B|\hat{s}_{j-1}=G))\\
&\qquad\left.+b_{j-1}\ln(\Pr(\hat{s}_j=B|\hat{s}_{j-1}=B))\right\}\\
&\geq\sum_{j=0}^{N-1}(1-b_j)\left\{\ln\left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)-\frac{(y_j-(-1)^{\tilde{v}_j}\sqrt{E_s})^2}{2\sigma^2}\right.\\
&\qquad+(1-b_{j-1})\ln(\Pr(\tilde{s}_j=G|\tilde{s}_{j-1}=G))\\
&\qquad\left.+b_{j-1}\ln(\Pr(\tilde{s}_j=G|\tilde{s}_{j-1}=B))\right\}\\
&+\sum_{j=0}^{N-1}b_j\left\{\ln\left(\frac{1}{\sqrt{2\pi R\sigma^2}}\right)-\frac{(y_j-(-1)^{\tilde{v}_j}\sqrt{E_s})^2}{2R\sigma^2}\right.\\
&\qquad+(1-b_{j-1})\ln(\Pr(\tilde{s}_j=B|\tilde{s}_{j-1}=G))\\
&\qquad\left.+b_{j-1}\ln(\Pr(\tilde{s}_j=B|\tilde{s}_{j-1}=B))\right\}\ .
\end{aligned}\tag{8}$$

As previously mentioned, because it is difficult to obtain the exact PDF of impulse noise in a bad channel state, the impulse statistics are considered unknown at the proposed decoder. Thus, when conditioned to a bad channel state, the decoder views the PDF as shown in (9), independently of y_j; thus, information regarding the bad channel state cannot be determined despite receiving y_j. Hence, $\Pr(y_j|\tilde{v}_j,\tilde{s}_j=B)$ is estimated as follows:

$$\Pr(y_j|\tilde{v}_j,\tilde{s}_j=B)=\frac{1}{\sqrt{2\pi R^{(d)}\sigma^2}}\ , \tag{9}$$

where $R^{(d)}$ is a design parameter that can be determined according to various channel parameters. Accordingly, regarding the proposed decoder, substituting (28) and (9) into (27) yields the proposed decoding rule:

$$\begin{aligned}
&\sum_{j=0}^{N-1}(1-b_j)\left\{\ln\left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)-\frac{(y_j-(-1)^{\hat{v}_j}\sqrt{E_s})^2}{2\sigma^2}\right.\\
&\qquad+(1-b_{j-1})\ln(\Pr(\hat{s}_j=G|\hat{s}_{j-1}=G))\\
&\qquad\left.+b_{j-1}\ln(\Pr(\hat{s}_j=G|\hat{s}_{j-1}=B))\right\}\\
&+\sum_{j=0}^{N-1}b_j\left\{\ln\left(\frac{1}{\sqrt{2\pi R^{(d)}\sigma^2}}\right)+(1-b_{j-1})\right.\\
&\qquad\cdot\ln(\Pr(\hat{s}_j=B|\hat{s}_{j-1}=G))\\
&\qquad\left.+b_{j-1}\ln(\Pr(\hat{s}_j=B|\hat{s}_{j-1}=B))\right\}\\
&\geq\sum_{j=0}^{N-1}(1-b_j)\left\{\ln\left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)-\frac{(y_j-(-1)^{\tilde{v}_j}\sqrt{E_s})^2}{2\sigma^2}\right.\\
&\qquad+(1-b_{j-1})\ln(\Pr(\tilde{s}_j=G|\tilde{s}_{j-1}=G))\\
&\qquad\left.+b_{j-1}\ln(\Pr(\tilde{s}_j=G|\tilde{s}_{j-1}=B))\right\}\\
&+\sum_{j=0}^{N-1}b_j\left\{\ln\left(\frac{1}{\sqrt{2\pi R^{(d)}\sigma^2}}\right)+(1-b_{j-1})\right.\\
&\qquad\cdot\ln(\Pr(\tilde{s}_j=B|\tilde{s}_{j-1}=G))\\
&\qquad\left.+b_{j-1}\ln(\Pr(\tilde{s}_j=B|\tilde{s}_{j-1}=B))\right\}\ .
\end{aligned}\tag{10}$$

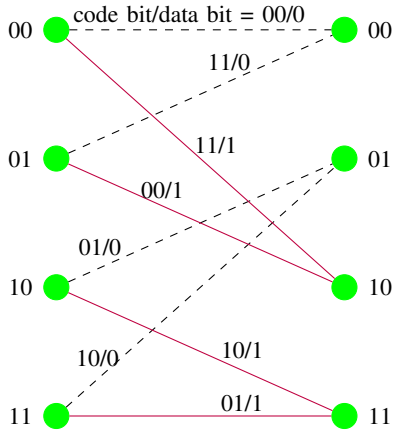
Based on (7), the decoding metric is clearly related not only to the code bit and its corresponding received bit, but also to the channel state. Based on the new decoding bit metric shown in (7), the VA was implemented using an expanded trellis diagram. The construction of the joint trellis diagram describes the channel and encoder states and each state in the encoder becomes a superstate containing all conceivable combinations of channel states; this is referred to as a *two-dimensional trellis diagram*. Add-compare-select (ACS) procedures [29], [30] were adopted to update the state metric. The state metric for the current trellis state depends on the predecessor state metric and the branch metrics between two trellis encoder states (the branch metric is added to the state metric of the previous time instant). Whenever numerous paths merge into one encoder state, only the most likely path is retained. However, unlike the traditional VA, in the proposed algorithm, two metric values are accumulated at each trellis state, corresponding to the good and bad channel states.

Fig. 3 shows an example that describes a two-dimensional trellis diagram for a (2,1,2) CC. In Fig. 3(b), the two circles enclosed by the external circle connote the two channel states (blue = good, red= bad). The filled and unfilled external circles represent the trellis encoder state and virtual state, respectively. Two branches (blue and red lines) are merged to each channel state (*G* and *B*), and the survivor candidate of the two branches is stored at the corresponding channel state of the virtual state. Similarly, four branches are merged to each channel state of the trellis encoder state. Thus, the most likely of the four solid lines is retained at channel state *G*, and the most likely path of the four dashed lines is retained at channel state *B* of the trellis encoder state. For every trellis encoder state, recursion yields an updated state metric, and at the final level of the trellis diagram, a comparison and decision must be made between accumulated metric values of the *G* and *B* channel states prior to tracing back the entire survivor path and decoding the information bits. Prior to introducing the recursive procedure of the proposed algorithm, $(n_{cc}-1)$ virtual states exist between the starting and ending states regarding the generalized framework of a (n_{cc},k_{cc},m) CC.

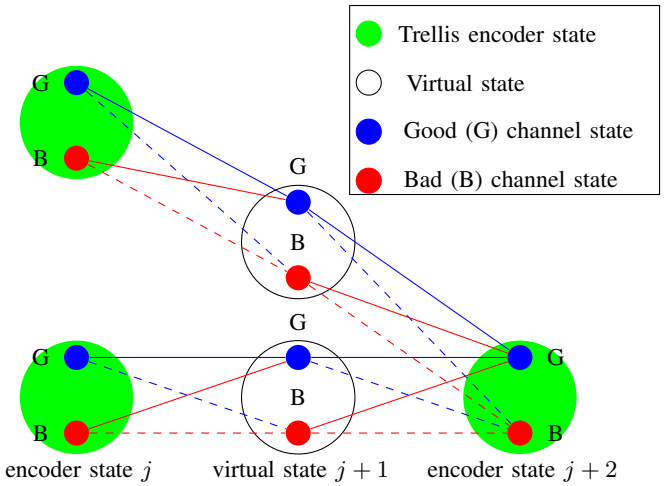
B. Metrics on Trellis Diagram

Let $x\in\{0,1,\ldots,2^m-1\}$ be the encoder state in the original trellis of a (n_{cc},k_{cc},m) CC. A node at level $(j+1)$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2014.2312727, IEEE Transactions on Vehicular Technology



(a)



(b)

Fig. 3. (a) Portion of trellis for (2,1,2) convolutional code (b) two-dimensional representation for parts of trellis states

of the expanded trellis is denoted by state vector $(\tilde{s}_j, x)$. The encoder state x of virtual state $(\tilde{s}_j, x)$ is the same as that of the encoder state of the first non virtual state tracing the branches back from this virtual state. In other words, every virtual state between two encoder states is labeled x, which is identical to the previous encoder state of the two encoder states. In the VA, the path metric value accumulated at state $(\tilde{s}_j, x)$ of the trellis diagram at the $(j+1)^{th}$ time index (at the $(j+1)^{th}$ level of the expanded trellis) is as follows:

$$\zeta_j^{(\tilde{s}_j,x)} = \max_{\tilde{\boldsymbol{v}}_0^j} \left\{ \Pr(\boldsymbol{y}_0^j | \tilde{\boldsymbol{v}}_0^j, \tilde{\boldsymbol{s}}_0^j) \Pr(\tilde{\boldsymbol{s}}_0^j) \right\} , \tag{11}$$

where $\tilde{\boldsymbol{v}}_0^j$ ends at state $(\tilde{s}_j, x)$ in the expanded trellis. In the VA, $\zeta_{j+1}^{(\tilde{s}_{j+1},x')}$ can be updated as a function of $\zeta_j^{(\tilde{s}_j,x)}$, $\tilde{v}_{j+1}$, $\tilde{s}_{j+1}$, and $\tilde{s}_j$, where $\tilde{v}_{j+1}$ corresponds to the branch between states $(\tilde{s}_j, x)$ and $(\tilde{s}_{j+1}, x')$. After calculating (24) and (11), the accumulated metric at state $(\tilde{s}_{j+1}, x')$ becomes the following:

$$\zeta_{j+1}^{(\tilde{s}_{j+1},x')} = \max_{\tilde{v}_{j+1}} \left\{ \zeta_j^{(\tilde{s}_j,x)} \left(\Pr(y_{j+1}|\tilde{v}_{j+1}, \tilde{s}_{j+1}) \Pr(\tilde{s}_{j+1}|\tilde{s}_j) \right) \right\} . \tag{12}$$

By calculating the logarithm of (28) and (9), the proposed decoding bit metric at the $(j+2)^{th}$ time can be directly expressed as follows:

$$\beta(y_{j+1}|\tilde{v}_{j+1}, \tilde{s}_{j+1}, \tilde{s}_j) = \ln[\Pr(y_{j+1}|\tilde{v}_{j+1}, \tilde{s}_{j+1})] + \ln[\Pr(\tilde{s}_{j+1}|\tilde{s}_j)]$$

$$= \begin{cases} \ln\left(\frac{\Pr(\tilde{s}_{j+1}=G|\tilde{s}_j)}{\sqrt{2\pi\sigma^2}}\right) - \frac{(y_{j+1}-(-1)^{\tilde{v}_{j+1}}\sqrt{E_s})^2}{2\sigma^2} & \text{if } \tilde{s}_{j+1} = G \\ \ln(\Pr(\tilde{s}_{j+1} = B|\tilde{s}_j)) + \ln\left(\frac{1}{\sqrt{2\pi R^{(d)}\sigma^2}}\right) & \text{if } \tilde{s}_{j+1} = B \end{cases} . \tag{13}$$

Therefore, depending on the previous and present states of the channel, four decoding metrics are possible. Because each state transition probability is unknown at the decoder, its assumed value is used; thus, when $\tilde{s}_{j+1} = G$, the two decoding metrics are

$$\beta(y_{j+1}|\tilde{v}_{j+1}, \tilde{s}_{j+1} = G, \tilde{s}_j = G) = \ln\left(\frac{P_{GG}^{(d)}}{\sqrt{2\pi\sigma^2}}\right) - \frac{(y_{j+1}-(-1)^{\tilde{v}_{j+1}}\sqrt{E_s})^2}{2\sigma^2} ,$$

and

$$\beta(y_{j+1}|\tilde{v}_{j+1}, \tilde{s}_{j+1} = G, \tilde{s}_j = B) = \ln\left(\frac{P_{BG}^{(d)}}{\sqrt{2\pi\sigma^2}}\right) - \frac{(y_{j+1}-(-1)^{\tilde{v}_{j+1}}\sqrt{E_s})^2}{2\sigma^2} .$$

Similarly, when $\tilde{s}_{j+1} = B$, the decoding metrics are

$$\beta(y_{j+1}|\tilde{v}_{j+1}, \tilde{s}_{j+1} = B, \tilde{s}_j = G) = \ln(P_{GB}^{(d)}) + \ln\left(\frac{1}{\sqrt{2\pi R^{(d)}\sigma^2}}\right) ,$$

and

$$\beta(y_{j+1}|\tilde{v}_{j+1}, \tilde{s}_{j+1} = B, \tilde{s}_j = B) = \ln(P_{BB}^{(d)}) + \ln\left(\frac{1}{\sqrt{2\pi R^{(d)}\sigma^2}}\right) .$$

As previously discussed, the intermediate stages for updating the accumulated metrics can be executed only after the decoding algorithm is initiated. Accordingly, the initial step path metric values of state $(\tilde{s}_0, x)$ regarding the initial state probability $\Pr(\tilde{s}_0)$, initial metric value of encoder state $\zeta_{-1}^{(\tilde{s}_{-1},x)}$, and noise PDF are calculated as follows:

$$\begin{aligned} \zeta_0^{(\tilde{s}_0,x)} &= \zeta_{-1}^{(\tilde{s}_{-1},x)} + \beta\left(y_0|\tilde{v}_0, \tilde{s}_0, \tilde{s}_{-1}\right) \\ &= \zeta_{-1}^{(\tilde{s}_{-1},x)} + \ln\left[\Pr(\tilde{s}_0)\right] + \ln\left[\Pr(y_0|\tilde{v}_0, \tilde{s}_0)\right] \\ &= \zeta_{-1}^{(\tilde{s}_{-1},x)} + \begin{cases} \ln\left(\frac{P_G^{(d)}}{\sqrt{2\pi\sigma^2}}\right) - \frac{(y_0-(-1)^{\tilde{v}_0}\sqrt{E_s})^2}{2\sigma^2} \\ \qquad \text{if} \quad \tilde{s}_0 = G \\ \ln(P_B^{(d)}) + \ln\left(\frac{1}{\sqrt{2\pi R^{(d)}\sigma^2}}\right) \\ \qquad \text{if} \quad \tilde{s}_0 = B \end{cases} , \end{aligned} \tag{14}$$

where $\Pr(\tilde{s}_0) \in \{P_B^{(d)}, P_G^{(d)}\}$ is unknown but assumed at the decoder. $\zeta_{-1}^{(\tilde{s}_{-1},x)}$ is initialized to zero for all $x \in$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2014.2312727, IEEE Transactions on Vehicular Technology

$\{0, 1, \ldots, 2^m - 1\}$, whereas state $x = 0$ of the first trellis level is considered the only initial state (i.e., $\zeta_{-1}^{(\tilde{s}_{-1},0)} = 0$ and $\zeta_{-1}^{(\tilde{s}_{-1},x)} = -\infty$ for all $x \in \{1, \ldots, 2^m - 1\}$) because the algorithm selects and retains the path that exhibits the maximal cumulative metric. In (14), the path metric value $\zeta_0^{(\tilde{s}_0,x)}$ is accumulated at the first virtual state of the trellis diagram. However, the steps in the expanded trellis diagram are followed to derive the general formula for the state metric. A crucial step in the VA is updating the state metric by using a recursive operation (i.e., the new values of the state metric are a function of the previous values). Thus, to update the accumulated or state metric at state vectors of the current trellis step, the VA requires the previously accumulated state metric value and the branch metrics between the states at various trellis steps.

To compute the recursive state metric update at the trellis step, various notations must be denoted. Fig. 4 shows the example of (2,1,2) CC (c.f., Fig. 3(a)); let the starting state at time j be $(\tilde{s}_{j-1}, x)$, the ending state at the $(j+2)^{th}$ time index be $(\tilde{s}_{j+1}, x')$, and the virtual state (at the $(j+1)^{th}$ time index) between the starting and ending states have the same encoder state notation as the starting state and be denoted by $(\tilde{s}_j, x)$. A description is subsequently provided in a general CC context.

Because the bit metric shown in (7) is a function of both the channel and encoder states and the transitions between the states determine the branches, the state transitions are incorporated with the code bits associated with these branches. In the trellis state metric update procedure, the difference between the virtual and trellis encoder states is the number of branches merging at each channel state because the branches merging at the trellis encoder state of the expanded trellis diagram originate from various paths. Thus, a general formula can be used to compute the state metric update.

Because the procedure for updating the state metric is the same for any code bit length, $n_{cc} = 2$ can be used to show the detail of the proposed metric. For every state $(\tilde{s}_{j+1}, x')$, a recursion yields the updated state metric $\zeta_{j+1}^{(\tilde{s}_{j+1},x')}$, which is based on the previous state metric $\zeta_j^{(\tilde{s}_j,x)}$ connected to $(\tilde{s}_{j+1}, x')$ and the current branch metric. The survivor state metric at each channel state is determined by computing the maximum value of all possible candidates that merge at state $(\tilde{s}_{j+1}, x')$. Thus, the updated state metric can be expressed as follows:

$$\zeta_{j+1}^{(\tilde{s}_{j+1},x')} = \max_{(\tilde{s}_j,x)} \left\{ \zeta_j^{(\tilde{s}_j,x)} + \beta(y_{j+1}|\tilde{v}_{j+1}, \tilde{s}_{j+1}, \tilde{s}_j) \right\}, \quad (15)$$

where $\tilde{v}_{j+1}$ determines the number of branches that diverge from all possible $(\tilde{s}_j, x)$ states and merge at the current state $(\tilde{s}_{j+1}, x')$.

The trellis diagram shown in Fig. 4 is used to elucidate the recursion procedure for the proposed decoder of rate-1/2 CC, where only one virtual state exists between the starting and ending states. Therefore, at the $(j+1)^{th}$ time index, $\zeta_j^{(\tilde{s}_j=G,x)}$ and $\zeta_j^{(\tilde{s}_j=B,x)}$ of the virtual state, which exhibit the same encoder state as the beginning state, are respectively

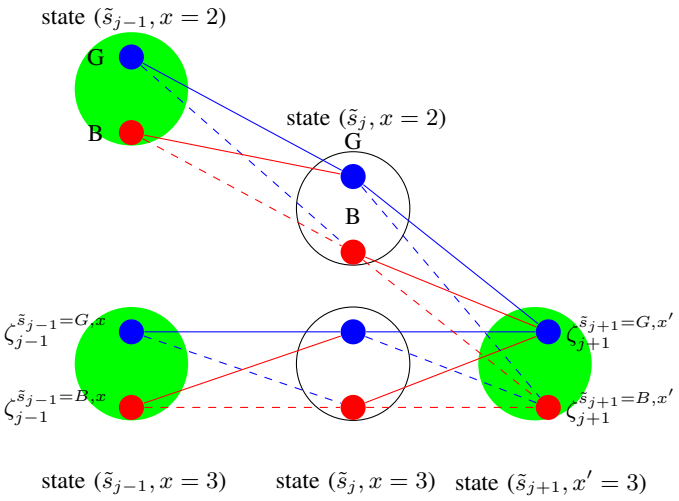


Fig. 4. Example of expanded trellis diagram to update path metric for $k_{cc} = 1, n_{cc} = 2$

determined as follows:

$$\zeta_j^{(\tilde{s}_j=G,x)} = \max_{(\tilde{s}_{j-1},x)} \left\{ \zeta_{j-1}^{(\tilde{s}_{j-1},x)} + \beta(y_j|\tilde{v}_j, \tilde{s}_j, \tilde{s}_{j-1}) \right\}$$

$$= \max \begin{cases} \zeta_{j-1}^{(\tilde{s}_{j-1}=G,x)} - \frac{(y_j - (-1)^{\tilde{v}_j}\sqrt{E_s})^2}{2\sigma^2} \\ \quad + \ln\left(\frac{P_{GG}^{(d)}}{\sqrt{2\pi\sigma^2}}\right) \quad \text{if} \quad \tilde{s}_{j-1} = G \\ \zeta_{j-1}^{(\tilde{s}_{j-1}=B,x)} - \frac{(y_j - (-1)^{\tilde{v}_j}\sqrt{E_s})^2}{2\sigma^2} \\ \quad + \ln\left(\frac{P_{BG}^{(d)}}{\sqrt{2\pi\sigma^2}}\right) \quad \text{if} \quad \tilde{s}_{j-1} = B \end{cases}, \quad (16)$$

and

$$\zeta_j^{(\tilde{s}_j=B,x)} = \max_{(\tilde{s}_{j-1},x)} \left\{ \zeta_{j-1}^{(\tilde{s}_{j-1},x)} + \beta(y_j|\tilde{v}_j, \tilde{s}_j, \tilde{s}_{j-1}) \right\}$$

$$= \max \begin{cases} \zeta_{j-1}^{(\tilde{s}_{j-1}=G,x)} + \ln\left(\frac{P_{GB}^{(d)}}{\sqrt{2\pi R^{(d)}\sigma^2}}\right) & \text{if } \tilde{s}_{j-1} = G \\ \zeta_{j-1}^{(\tilde{s}_{j-1}=B,x)} + \ln\left(\frac{P_{BB}^{(d)}}{\sqrt{2\pi R^{(d)}\sigma^2}}\right) & \text{if } \tilde{s}_{j-1} = B \end{cases}. \quad (17)$$

At the ending state $(\tilde{s}_{j+(n_{cc}-1)}, x') = (\tilde{s}_{j+1}, x')$ of the trellis, $2^{k_{cc}} = 2$ paths from two encoder states merge at the current encoder state. Therefore, using possible paths from state $(\tilde{s}_{j-1}, x)$ to $(\tilde{s}_{j+1}, x')$, $\zeta_{j+1}^{(\tilde{s}_{j+1}=G,x')}$, and $\zeta_{j+1}^{(\tilde{s}_{j+1}=B,x')}$ can be directly determined using (15).

When considering a general case (i.e., (n_{cc}, k_{cc}, m) CC), let the starting and ending states be $(\tilde{s}_{j-1}, x)$ and $(\tilde{s}_{j+(n_{cc}-1)}, x')$, respectively, where $x, x' \in \{0, 1, 2, \ldots, 2^m - 1\}$ and $\tilde{s}_{j-1}, \tilde{s}_{j+(n_{cc}-1)} \in \{G, B\}$. Furthermore, denote the $(n_{cc} - 1)$ virtual states between $(\tilde{s}_{j-1}, x)$ and $(\tilde{s}_{j+(n_{cc}-1)}, x')$ as $\big((\tilde{s}_j, x), (\tilde{s}_{j+1}, x), \ldots, (\tilde{s}_{j+(n_{cc}-2)}, x)\big)$, where $\tilde{s}_j, \tilde{s}_{j+1}, \ldots, \tilde{s}_{j+(n_{cc}-2)} \in \{G, B\}$, and, regardless of the channel state, these $(n_{cc} - 1)$ virtual states all exhibit the same encoder state x as the starting state. The corresponding state metrics $\left(\zeta_j^{(\tilde{s}_j,x)}, \zeta_{j+1}^{(\tilde{s}_{j+1},x)}, \ldots, \zeta_{j+(n_{cc}-2)}^{(\tilde{s}_{j+(n_{cc}-2)},x)}\right)$, which are based on the condition of the channel state, can be recursively measured using (16) and (17), respectively,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2014.2312727, IEEE Transactions on Vehicular Technology

to assess the good and bad channel states. Examining the state transition between consecutive encoder states ($2^{k_{cc}}$ paths originated from $2^{k_{cc}}$ encoder states and merged at the ending state $(\tilde{s}_{j+(n_{cc}-1)}, x')$) indicates that the updated state metrics, namely $\zeta_{j+(n_{cc}-1)}^{(\tilde{s}_{j+(n_{cc}-1)}=G,x')}$ and $\zeta_{j+(n_{cc}-1)}^{(\tilde{s}_{j+(n_{cc}-1)}=B,x')}$, can be computed using (15).

C. Computational Complexity

The decoder speed and memory use are essential gauges of complexity [30]. The number of operations involving both states and branches governs the memory storage and computational requirements of the VA. The trellis state space dimension is used to measure the state complexity, whereas the total number of branches is used to measure the branch complexity. The space complexity of an algorithm is the number of storage elements that must be reserved for its use, whereas the time complexity counts the number of arithmetic operations, and an operation can be an addition or comparison. Complexity of the Viterbi decoding depends on the number of metric computations between successive states of the trellis diagram because the computational complexity is measured by the number of branch metric calculations. Thus, the time required to process the number of branches in the ACS circuit and the total branches in the trellis govern the decoding speed [31], and the number of arithmetic operations is required to determine the complexity of the proposed algorithm.

To determine the survivor path for the trellis encoder states at each step of the trellis diagram, the proposed algorithm performs ACS operations. Hence, the number of ACS operations per trellis step and the number of additions and comparisons involved in the ACS operation dictate the computational complexity of this algorithm, explaining the complexity of computing ACS operations. In an ACS operation, the metric of the survivor from which the branch diverges is added to the branch metrics; all metrics along the paths that converge to each channel state are compared and the path that exhibits the maximal path metric value is selected and preserved. The total number of addition operations is considered to compute the complexity of the proposed algorithm. The additions are the arithmetic operations used to calculate the branch metrics, and the operation used to add the branch and state metric values of the predecessor trellis encoder states.

To generalize the computational complexity of the proposed algorithm for a (n_{cc}, k_{cc}, m) CC, the number of arithmetic operations is compared with that of the traditional VA. Regarding a traditional VA, $n_{cc} \times 2^{k_{cc}}$ addition operations are conducted at each trellis state because the number of code bits between two trellis levels of a single path is n_{cc}, and $2^{k_{cc}}$ paths are merged at each trellis state. This indicates that on each path, $(n_{cc}-1)$ addition operations are required for each branch metric computation [32] and one additional operation is required to add the branch and state metric values. Given a number of 2^m trellis states, the total number of addition operations between two trellis levels is $2^m \times n_{cc} \times 2^{k_{cc}} = n_{cc} \times 2^{m+k_{cc}}$. Furthermore, because $2^{k_{cc}} - 1$ comparisons are involved in each trellis state, the number of comparisons required to determine the survivor path of the metrics for all 2^m states at each trellis level is $2^m \times (2^{k_{cc}} - 1)$ [30]. Thus, the computational complexity of a traditional VA is expressed as follows:

$$\begin{aligned} &2^m \times \left(\left(n_{cc} \times 2^{k_{cc}}\right) + \left(2^{k_{cc}} - 1\right)\right) \\ &= 2^m \times \left(2^{k_{cc}}\left(n_{cc}+1\right) - 1\right) \approx \left(n_{cc}+1\right) \times 2^{k_{cc}+m}. \end{aligned} \quad (18)$$

If the addition operations for branch metric computation are not considered, the computational complexity is $2^m \times (2^{k_{cc}} + (2^{k_{cc}} - 1)) \approx 2^{k_{cc}+m+1} = 2^{k_{cc}+v_{cc}}$ [33], where v_{cc} is the constraint length. However, in this study, additional operations were considered when computing the branch metrics.

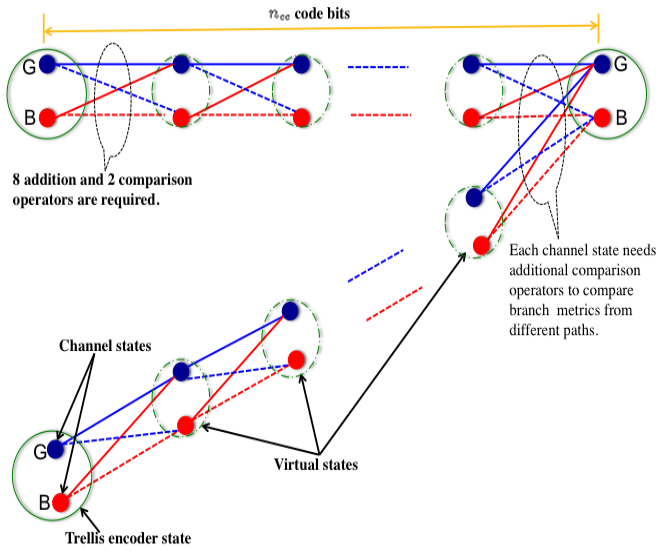


Fig. 5. Trellis diagram between two trellis encoder levels of n_{cc} code bits

The state metrics calculated in (15), (16), and (17) for $n_{cc} = 2$ can be generalized to any $n_{cc} \geq 2$, as depicted in Fig. 5. When the number of trellis encoder states equals 2^m, and each encoder state encompasses two channel states, the total number of states equals $2 \times 2^m = 2^{m+1}$. Because various branches (that diverged from good and bad channel states) merge at the virtual states, the number of addition and comparison operations is calculated stepwise throughout the virtual states and the trellis encoder states of the trellis. The first step is to determine the number of addition operations between two trellis levels.

As shown in Fig. 5, two branches (solid and dashed lines) diverged from each channel state and two branches merged at each channel state (blue and red solid lines merge to the good channel state, and blue and red dashed lines merge to the bad channel state). Each branch involves two additions; therefore, eight addition operations are counted between the first trellis encoder state and the first virtual state. Similarly eight addition operations are required between every two virtual states, and the same number of addition operations are required between the last virtual state and the last trellis encoder state. Therefore, after accounting for n_{cc} code bits on each path between the two trellis levels, the number of addition operations in a single path at each trellis encoder state is $8n_{cc}$. Furthermore, because $2^{k_{cc}}$ paths (when both good and bad states are combined into

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2014.2312727, IEEE Transactions on Vehicular Technology

one encoder state) are merged at each trellis encoder state, and 2^m encoder states are involved at each trellis level, the number of addition operations between the two trellis levels is calculated as follows:

$$8n_{cc} \times 2^{k_{cc}} \times 2^m = n_{cc} \times 2^{k_{cc}+m+3} \ . \quad (19)$$

Similar to the previous derivation, the number of comparison operations involved is calculated. As shown in Fig. 5, two comparison operations occur between each trellis encoder state and virtual state, and two comparisons occur between every two virtual states, where the two branches merged to the good channel state count as the first comparison, and the branches merged to the bad channel state count as the second comparison. Given n_{cc} code bits on any single path between two trellis steps, $2n_{cc}$ comparison operations are used. When counting $2^{k_{cc}}$ paths (both good and bad states are combined into one encoder state), the number of comparison operations is $2n_{cc} \times 2^{k_{cc}} = n_{cc} \times 2^{k_{cc}+1}$. Moreover, Fig. 5 shows that during each channel state metric value accumulation procedure, additional comparison operators are required to compare the path metrics originating from distinct paths. The comparison formula in [30] was adopted for channel states such that $2^{k_{cc}}-1$ comparisons were added for each of channel state yielding $2\times(2^{k_{cc}}-1)$ additional comparison operators at each trellis encoder state. Accounting for 2^m encoder states, the general formula for calculating the number of comparisons between two trellis levels is as follows:

$$2^m \times \left(n_{cc} \times 2^{k_{cc}+1} + 2 \times \left(2^{k_{cc}} - 1\right)\right) = 2^{m+1}\left(2^{k_{cc}}\left(n_{cc}+1\right)-1\right). \quad (20)$$

Combining (19) and (20), the computational complexity of the proposed decoder is expressed as follows:

$$\begin{aligned}\left(n_{cc} \times 2^{k_{cc}+m+3}\right) + 2^{m+1}\left(2^{k_{cc}}\left(n_{cc}+1\right)-1\right) &= 2^{m+1}\left(2^{k_{cc}} \times (5n_{cc}+1)-1\right) \\ \approx 2^{m+1}\left(2^{k_{cc}} \times (5n_{cc}+1)\right) &= (10n_{cc}+2) \times 2^{m+k_{cc}} \ . \quad (21)\end{aligned}$$

Comparing (21) with the computational complexity of the VA calculated in (18) indicates that the proposed decoder yields a complexity increase of $\frac{10n_{cc}+2}{n_{cc}+1}$.

Moreover, the computational complexities of the Viterbi and proposed algorithms were numerically evaluated using various memory lengths, m and the number of k_{cc}-size data sub$-$blocks, namely L (the length of an information bit sequence is $k_{cc}L$). The results are listed in Table I; the proposed algorithm slightly increased the complexity compared with the traditional VA, regardless of the selected code parameters m and L.

TABLE I
NUMERICAL COMPUTATIONAL COMPLEXITY COMPARISON

(n_{cc}, k_{cc}, m)	L	Computational complexity	
		Traditional Viterbi algorithm	Proposed algorithm
$(2,1,6)$	100	4.07×10^4	2.98×10^5
$(2,1,6)$	500	1.94×10^5	1.42×10^6
$(2,1,20)$	100	7.55×10^8	5.54×10^9
$(2,1,20)$	500	3.27×10^9	2.40×10^{10}

IV. SIMULATION RESULTS

The proposed decoding algorithm was assessed using computer simulations of bit error rate (BER) versus signal-to-noise ratio (SNR) (i.e., E_b/N_0 in dB, where E_b is the energy of an information bit). A (2,1,6) *CC* with generators 147, 135 (octal) was used and the number of information bits per data frame was fixed at $L = 500$. The interleaver depth I was set at 20 unless otherwise specified. The performance of our proposed algorithm was investigated in single carrier communication schemes using the BPSK modulation under the Markov Gaussian channel. Because the transition probability in the memory channel is unavailable to the proposed decoder, a person-by-person optimization search [34] was invoked to estimate P_{BG} values in various scenarios. Although the proposed decoder is robust against P_{BG} values, the $P_{BG}^{(d)}$ value is obtained by testing for possible ranges of its values at a fixed SNR value and an interleaver depth of 20. The other transition probabilities can be determined using (4) after η^I is calculated using this $P_{BG}^{(d)}$ value. In this regard, $P_{BG}^{(d)} = 0.035$ was obtained and used in the corresponding simulations. The optimal decoder referred to the decoder that had full access to the impulse statistics when determining the maximum likely transmitted codeword among the set of possible codewords.

The alpha-PFD [13], [20] has been proposed for decoding CCs through impulse noise channels; hence, it was compared with the proposed scheme in the simulations. The alpha-PFD was used to suppress sample y_j at an extremely large magnitude (likely to be corrupted by impulse) by using the following penalty function [35]

$$\beta(\tilde{v}_j \mid y_j, \alpha) = \frac{1}{2\alpha} \exp\left(-\alpha(y_j - (-1)^{\tilde{v}_j}\sqrt{E_s})^2\right) \ , \alpha > 0, \quad (22)$$

as the bit metric, rather than the Euclidean distance, namely $-(y_j - (-1)^{\tilde{v}_j}\sqrt{E_s})^2$, which is widely used in the VA for codeword estimation in the AWGN channel. When $\alpha \to 0$, (22) was approximated to the Euclidean distance metric. Except for the exponential function and multiplicative weight, the computational complexity of implementing the alpha-PFD by using metric (22) on the trellis tends to be identical to that of the traditional VA. Moreover, the suitable range of α for the alpha-PFD was analytically determined and justified using simulations in [20]. This section demonstrates that the alpha-PFD, which does not exploit the memory properties of the impulse noise channel, experiences performance loss because of its simplified computations.

Simulation results were reported by testing the decoders at fairly wide SNR range as well as various interleaver depths. As pointed out in [20], it is plausible to configure the interleaver depth to be twice the average burst length in major practical scenarios. In light of this, most of the following BER curves were obtained in the context when an interleaver depth was set at $I = \frac{B_L}{2} = 20$. Furthermore, because, in reality, communication systems target the BER at 10^{-5}, all of the simulations were conducted such that the derived results would encompass that critical point, regardless of system configurations used.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2014.2312727, IEEE Transactions on Vehicular Technology

A. Infinite Interleaving

The BER performance levels of various decoding schemes were examined, using multiple channel parameters such as $P_B = 0.025$, $P_G = 0.975$, $\gamma = 1$, and $R = 100$ as shown in Fig. 6. Because of a lack of statistical impulse knowledge, the proposed decoder assumed $P_B^{(d)} = 0.01$ and $R^{(d)} = 200$. The optimal $\alpha = 0.5$ for the employed *CC* was assumed in the alpha-PFD metric [20]. Fig. 6 shows that the BER performance of the proposed scheme was similar to that of the optimal decoder. By contrast, the alpha-PFD metric associated with the BER curve labeled by the dash-line marked "$\triangleright$", induced approximately 0.5 dB of performance loss. Because the interleaver depth I is sufficiently large relative to the burst lengths of the impulses, $1/P_{BG}$, the real-time restrictions on communication were likely to be relaxed.

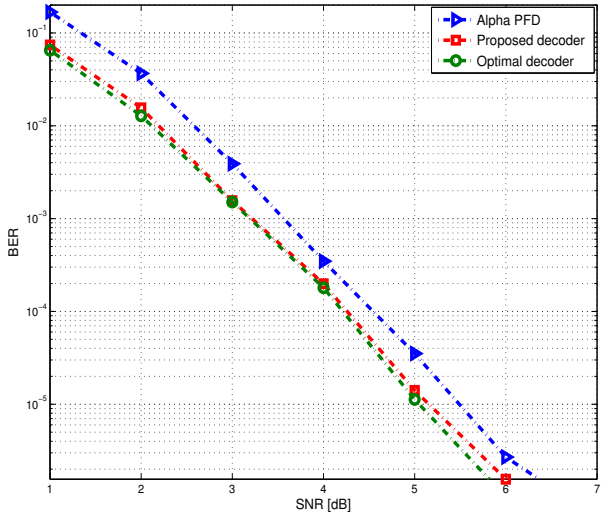


Fig. 6. Comparison of the BER performance for different decoding schemes with channel parameters $P_B = 0.025$, and R=100, assuming I is extremely large relative to $1/P_{BG}$. The alpha-penalty function decoder (alpha-PFD) uses (22) as the penalty function, whereas the optimal decoder, assuming the statistical knowledge of the impulse noise, uses the conditional PDF (29) along with transition probabilities for the MAP decoding.

B. Finite Interleaving

Finite interleaving was assumed for the remaining simulations, and various scenarios were established to present the robustness of the proposed decoder.

1) Robustness against R and P_{BG} for various P_B: Robustness of the proposed decoder was investigated for assumed R and P_{BG} values at various impulse occurrence probabilities. The BER results shown in Figs. 7, 8, and 9 corresponded to $P_B = 0.01$, 0.025, and 0.04, respectively. The relative strength between the impulses and AWGN was fixed at $R = 100$, and a transition probability of $P_{BG} = 0.025$. The proposed decoder assumed that P_B was known in the present setting; that is $P_B^{(d)} = P_B$, while the estimate of R (i.e., $(R^{(d)})$) was set at 200, and $P_{BG}^{(d)} = 0.035$. The burst length deviated from the actual value of 40, because $P_{BG}^{(d)} \neq P_{BG}$.

Figs. 7, 8, and 9 show that the proposed decoder performed similarly to the optimal decoder, regardless of the probability

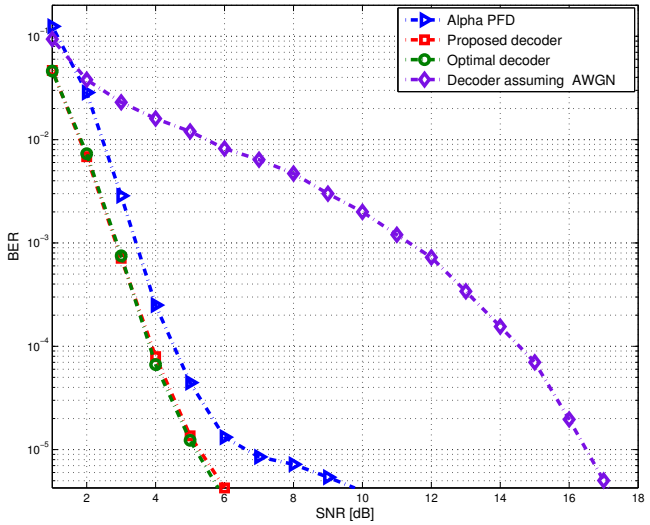


Fig. 7. BER performance in Markov Gaussian channel with channel parameters $P_B = 0.01$ and $R = 100$.

of impulse occurrence P_B. As the P_B increased, the gap in the level of BER performance widened such that the proposed decoding scheme outperformed the alpha-PFD; a gap wider than 4 dB SNR was observed when $P_B = 0.04$ at the BER of 10^{-5} (c.f., Fig. 9), implying that the proposed decoding scheme was robust against model mismatches arising from unknowns such as R and P_{BG}.

Fig. 7 shows that the simulation results were derived from four decoders in the Markov Gaussian channel, using parameters of $P_B = 0.01$ and $R = 100$. The curve labeled 'Decoder assuming AWGN' shows the performance level attained using the conventional Euclidean bit metric $-(y_j - (-1)^{\tilde{v}_j}\sqrt{E_s})^2$, demonstrating that the impulsive effect was completely neglected at the receiver. Notably, the proposed decoder yielded a BER performance level similar to that of the optimal decoder, which required statistics for the memory impulse noise channel; however, the BER of the decoder that neglected the impulsive effect (see the dash-dot line marked with '$\diamond$') did not descend to 10^{-5} until the SNR was 17 dB, demonstrating an 11 dB performance loss compared with our proposed decoder.

2) Effect of the interleaver length on BER performance: As previously mentioned, the BER performance deteriorated as the P_B value increased. To counter the negative influence of the large P_B, the effect of the interleaver depth on the BER was examined; however, the considered framework was not equipped with a feedback loop to leverage the interleaver depth at the transmitter. Assuming a simulation setup identical to that shown in Fig. 9 (except for the interleaver depth), Fig. 10 shows the BER performance level when length $I = 30$. The proposed scheme and optimal decoder yielded similar levels of performance regarding the BER curves (labeled by dash-dot lines) which continually declined. As a result, the BER performance of the proposed decoder was significantly improved when the interleaver depth was increased from $I = 20$ to $I = 30$ (c.f., Fig. 9 and Fig. 10).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2014.2312727, IEEE Transactions on Vehicular Technology

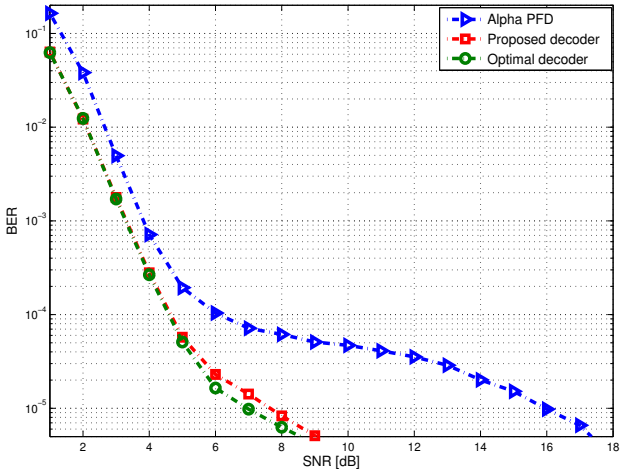


Fig. 8. BER performance in Markov Gaussian channel with channel parameters $P_B = 0.025$ and $R = 100$.

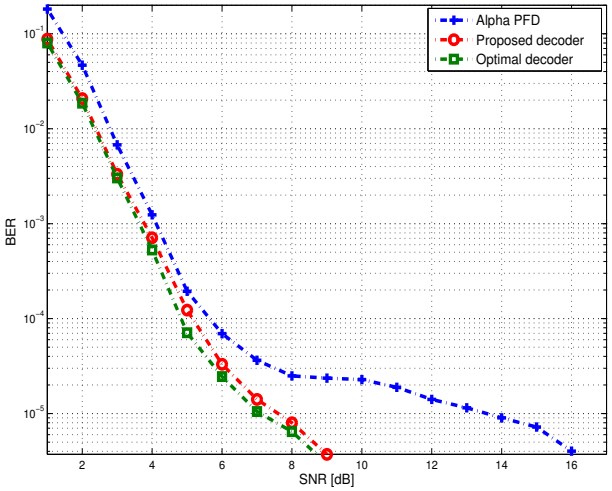


Fig. 10. BER performance in Markov Gaussian channel with channel parameters $P_B = 0.04$ and $R = 100$, and interleaver depth $I = 30$.

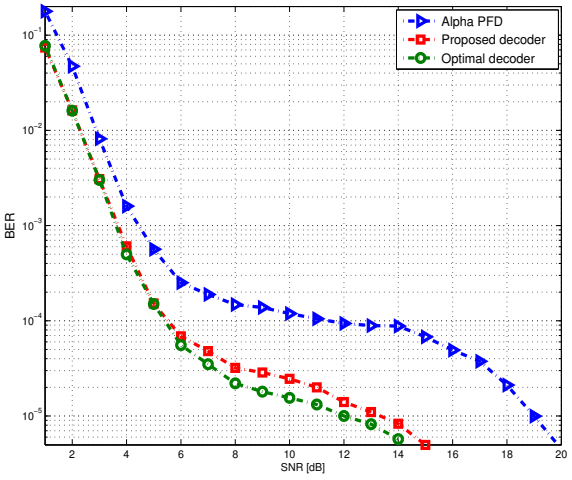


Fig. 9. BER performance in Markov Gaussian channel with channel parameters $P_B = 0.04$ and $R = 100$.

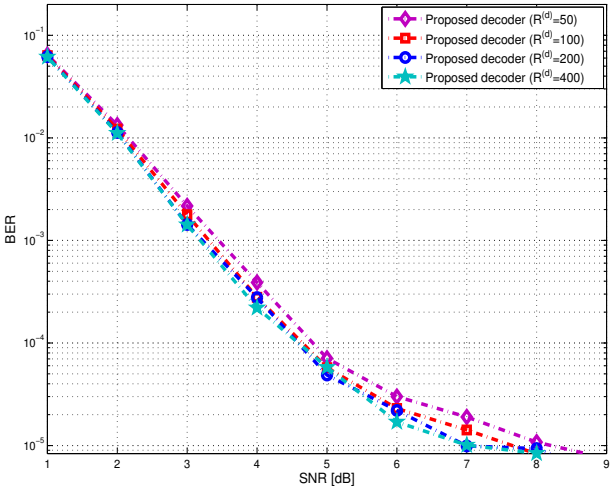


Fig. 11. BER performance for the proposed decoder with different guesses of R ($R^{(d)}$=50, 100, 200, and 400) in Markov Gaussian channel, for which $P_B = 0.025$, $P_{BG} = 0.025$, and $R = 100$.

3) Effect of the estimated impulse strength on BER performance: It was evaluated how the assumed average power ratios of the noise between the bad and good channel states, $R^{(d)}$, affected the BER performance level, examining the proposed decoder over a Markov Gaussian channel, for which $P_B = 0.025$, $P_{BG} = 0.025$, and $R = 100$. Fig. 11 shows the performance of the proposed decoder for various assumed impulse strength values with $P_B^{(d)}$ at the decoder different from the actual value P_B. Compellingly, the BER performance was not varied with the choice of $R^{(d)}$ though a slight performance loss was observed only when $R^{(d)} = 50$.

4) Robustness of the proposed scheme in a Markov Gaussian channel: A robustness inspection was conducted to examine the proposed decoding algorithm in the Markov Gaussian channel, for which $P_B = 0.025$, $P_{BG} = 0.025$, and $R = 100$; the multiple channel parameters were assumed to be unavailable to the decoder such that $R^{(d)} = 200$, $P_{BG}^{(d)} = 0.035$, and various assumed impulse occurrence probability values were considered instead. As shown in Fig. 12, the proposed decoder was remarkably robust at various assumed values of impulse occurrence probability: $P_B^{(d)} = 0.05$ and 0.1 were the two most exaggerated estimated values, inducing 1dB of SNR loss at BER $= 10^{-5}$.

The following are offered based on the simulation results to emphasize the effectiveness of the proposed decoder, which incorporates the (unknown) channel state transition probabilities. Recall that, in an ideal interleaving channel environment, Fig. 6 showed the alpha-PFD induced mere 0.5 dB SNR loss to our proposed decoder. However, when the finite interleaving constraint was enforced, our proposed decoder tremendously

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2014.2312727, IEEE Transactions on Vehicular Technology

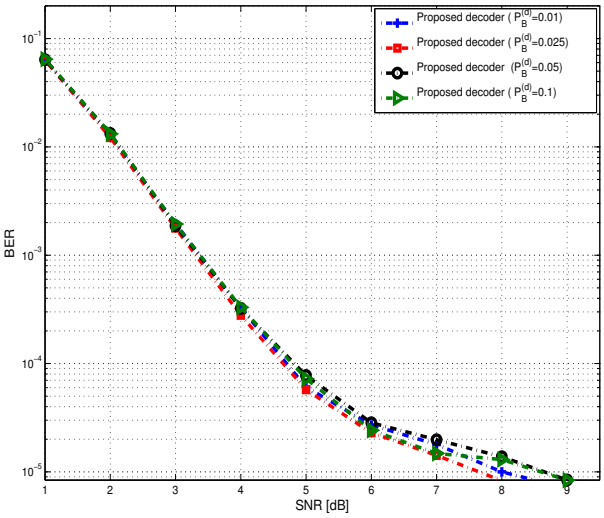


Fig. 12. BER performance for the proposed decoder assuming various probabilities of impulse occurrence ($P_B^{(d)} = 0.01, 0.025, 0.05$, and 0.1), and $R^{(d)} = 200$ over Markov Gaussian channel, for which $P_B = 0.025$, $P_{BG} = 0.025$, and $R = 100$.

outperformed the alpha-PFD because the inherent (unknown) channel memory was neglected not exploited; Fig. 8 shows that 8 dB SNR loss occurred at BER = 10^{-5} when comparing the alpha-PFD with the proposed decoder. As shown in Figs. 8, 9 and 10, the error floor encountered at high SNR values (when using the alpha-PFD metric) leads to performance gap, which is progressively widened with an increased probability of occurrence of impulses P_B. This is because the number of bursty impulses within a codeword of fixed length (i.e., N) increases with P_B, indicating that the (memoryless) channel model assumed in the alpha-PFD substantially differs from the encountered (Markov Gaussian) channel at large P_B values. When the SNR increases to a certain degree, the frequent impulse noise, which appears to govern the performance level, cannot be solely combatted by employing the VA without exploiting its inherent memory. The BER does not notably decline until the SNR value is excessively large (i.e., extremely small σ^2), a benign scenario likely to be free of strong impulses (recall that $\sigma_I^2 = (R-1)\sigma^2$).

V. Conclusion

An efficient and robust decoding approach was proposed to address impulse noises, characterized by a strong power in relation to the background noise and the common bursts that a generic communication systems are likely to be susceptible to. The efficiency of the proposed decoder was corroborated by enabling a VA like implementation and examining its computational complexity. The simulation results confirmed the robustness of the proposed decoding scheme: in general circumstances, the proposed decoder, which does not use impulse statistics, yielded a BER performance remarkably similar to that of an optimal decoder, requiring statistical impulse knowledge.

References

[1] K. Blackard, T. Rappaport, and C. Bostian, "Measurements and models of radio frequency impulsive noise for indoor wireless communications," *IEEE J. Sel. Areas Commun.*, vol. 11, no. 7, pp. 991–1001, Sep. 1993.

[2] H. Gu, I. Song, S. Yoon, and Y. Hee, "A class of spectrum-sensing schemes for cognitive radio under impulsive noise circumstances: Structure and performance in non-fading and fading environments," *IEEE Trans. Veh. Technol.*, vol. 59, no. 9, pp. 4322–4339, Nov. 2010.

[3] H. Meng, Y. Guan, and S. Chen, "Modeling and analysis of noise effects on broadband power-line communications," *IEEE Trans. Power Delivery*, vol. 20, no. 2, pp. 630–637, April 2005.

[4] C. Tepedelenlioglu and P. Gao, "On diversity reception over fading channels with impulsive noise," *IEEE Trans. Veh. Technol.*, vol. 54, no. 6, pp. 2037–2047, Nov. 2005.

[5] D. Middleton, "Canonical and quasi-canonical probability models of class A interference," *IEEE Trans. Electromagn. Compat.*, vol. 25, no. 2, pp. 76–106, May 1983.

[6] D. Middleton, "Statistical-physical models of electromagnetic interference," *IEEE Trans. Electromagn. Compat.*, vol. 19, no. 3, pp. 106–127, Aug. 1977.

[7] M. Ghosh, "Analysis of the effect of impulse noise on multicarrier and single carrier QAM systems," *IEEE Trans. Commun.*, vol. 44, no. 2, pp. 145–147, Feb. 1996.

[8] T. Trump and I. Muursepp, "An energy detector for spectrum sensing in impulsive noise environment," in *Proc. IEEE 22nd Int. Symp. on Personal, Indoor and Mobile Radio Commun.*, Sept. 2011, pp. 467–471.

[9] M. Nassar, P. Schniter, and B. L. Evans, "A factor graph approach to joint OFDM channel estimation and decoding in impulsive noise environments," *arXiv:1306.1851*, June 2013.

[10] D.-F. Tseng, Y. Han, W.-H. Mow, L.-C. Chang, and A.-J. Vinck, "Robust clipping for OFDM transmissions over memoryless impulsive noise channels," *IEEE Commun. Letters*, vol. 16, no. 7, pp. 1110–1113, July 2012.

[11] J. Lee and C. Tepedelenlioglu, "Space-time coding over fading channels with stable noise," *IEEE Trans. Veh. Technol.*, vol. 60, no. 7, pp. 3169–3177, Sept. 2011.

[12] D.-F. Tseng, Y. Han, W.-H. Mow, P.-N. Chen, J. Deng, and A.-J. Vinck, "Robust decoding for convolutionally coded systems impaired by memoryless impulsive noise," *IEEE Trans. Commun.*, vol. 61, no. 11, pp. 4640–4652, Sep. 2013.

[13] J. Mitra and L. Lampe, "Robust decoding for channels with impulse noise," in *Proc. IEEE Global Telecommun. Conf.*, Dec. 2006, pp. 1–6.

[14] R. Pighi, M. Franceschini, G. Ferrari, and R. Raheli, "Fundamental performance limits of communications systems impaired by impulse noise," *IEEE Trans. Commun.*, vol. 57, no. 1, pp. 171–182, Jan. 2009.

[15] D. Fertonani and G. Colavolpe, "A simplified metric for soft-output detection in the presence of impulse noise," in *Proc. IEEE International Symp. on Power Line Communications and Its Applications*, Mar. 2007, pp. 121–126.

[16] X. Wang and H. Poor, "Robust multiuser detection in non-Gaussian channels," *IEEE Trans. Signal Process.*, vol. 47, no. 2, pp. 289–305, Feb. 1999.

[17] S. Miyamoto, M. Katayama, and N. Morinaga, "Performance analysis of QAM systems under class A impulsive noise environment," *IEEE Trans. Electromagn. Compat.*, vol. 37, no. 2, pp. 260–267, May 1995.

[18] M. Mushkin and I. Bar-David, "Capacity and coding for the Gilbert-Elliott channels," *IEEE Trans. Inf. Theory*, vol. 35, no. 6, pp. 1277–1290, Nov. 1989.

[19] D. Fertonani and G. Colavolpe, "On reliable communications over channels impaired by bursty impulse noise," *IEEE Trans. Commun.*, vol. 57, no. 7, pp. 2024–2030, July 2009.

[20] J. Mitra and L. Lampe, "Convolutionally coded transmission over Markov-Gaussian channels: Analysis and decoding metrics," *IEEE Trans. Commun.*, vol. 58, no. 7, pp. 1939–1949, July 2010.

[21] T. Li, W.-H. Mow, V. K. N. Lau, M. Siu, R. Cheng, and R. Murch, "Robust joint interference detection and decoding for OFDM-based cognitive radio systems with unknown interference," *IEEE J. Sel. Areas Commun.*, vol. 25, no. 3, pp. 566–575, Apr. 2007.

[22] J. Mitra and L. Lampe, "Coded narrowband transmission over noisy powerline channels," in *Proc. IEEE International Symp. on Power Line Communications and Its Applications*, Apr. 2009, pp. 143–148.

[23] S. Haykin, "Cognitive radio: Brain-empowered wireless communications," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 2, pp. 201–220, Feb. 2005.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2014.2312727, IEEE Transactions on Vehicular Technology

[24] Z. Chen, C.-X. Wang, X. Hong, J. Thompson, S. Vorobyov, and X. Ge, "Interference modeling for cognitive radio networks with power or contention control," in *Proc. IEEE Wireless Commun. and Network. Conf. (WCNC)*, Apr. 2010, pp. 1–6.
[25] P. Pinto and M. Win, "Communication in a Poisson field of interferers," in *Proc. 40th Annual Conference on Information Sciences and Systems*, Mar. 2006, pp. 432–437.
[26] G. Yue, "Antijamming coding techniques," *IEEE Signal Process Mag.*, vol. 25, no. 6, pp. 35–45, 2008.
[27] F. Moghimi, A. Nasri, and R. Schober, "Adaptive L_p-norm spectrum sensing for cognitive radio networks," *IEEE Trans. Commun.*, vol. 59, no. 7, pp. 1934–1945, July 2011.
[28] M. Zimmermann and K. Dostert, "Analysis and modeling of impulsive noise in broad-band powerline communications," *IEEE Trans. Electromagn. Compat.*, vol. 44, no. 1, pp. 249–258, Feb. 2002.
[29] A. Viterbi and J.Omura, *Principles of Digital Communication and Coding*. New York: McGrew-Hill, 1979.
[30] S. Lin and D. J. C. Jr., *Error Control Coding: Fundamentals and Applications*. Englewood Cliffs, NJ: Prentice Hall Inc., 2004.
[31] M. Fossorier and S. Lin, "Differential trellis decoding of convolutional codes," *IEEE Trans. Inf. Theory*, vol. 46, no. 3, pp. 1046–1053, May 2000.
[32] H.-L. Lou, "Implementing the viterbi algorithm," *IEEE Signal Process. Mag.*, vol. 12, no. 5, pp. 42–52, Sep. 1995.
[33] G. Pottie and D. Taylor, "A comparison of reduced complexity decoding algorithms for trellis codes," *IEEE J. Sel. Areas Commun.*, vol. 7, no. 9, pp. 1369–1380, Dec. 1989.
[34] D. Bauso and R. Pesenti, "Generalized person-by-person optimization in team problems with binary decisions," in *Proc. IEEE American Control Conf.*, June 2008, pp. 717–722.
[35] B. Seyfe and S. Valaee, "A new choice of penalty function for robust multiuser detection based on M-estimation," *IEEE Trans. Commun.*, vol. 53, no. 2, pp. 224–227, Mar. 2005.

Fikreselam Gared Mengistu received B.Sc. in Electrical Engineering from Bahirdar University faculty of Technology, Bahirdar, Ethiopia in 2006, and M.Sc in Communication Engineering from Addis Ababa University Institute of Technology, Addis Ababa, Ethiopia in 2011. From March 1999 to January 2001 he worked as transmission technician in Ethiopia Telecommunication Corporation. He was Graduate assistant II, Assistant lecturer, and lecturer from July 2006 to January 2011 in School of computer and Electrical Engineering, Institute of Technology of Bahirdar University, Bahirdar, Ethiopia. He is currently a Ph.D. candidate at National Taiwan University of Science and Technology (NTUST), Taipei, Taiwan. His research interests are in the areas of decoding algorithms over Markov Gaussian channel.

Der-Feng Tseng received his B.S.E.E. degree from National Chiao Tung University, and M.S.E.E. and Ph.D. degrees from Purdue University, West Lafayette, Indiana. He was with TRW-ESL, Sunnyvale, CA, and Motorola Mobility, Pistcataway, NJ, as Senior Systems Engineer engaged in embedded software development for mobile devices before joining National Taiwan University of Science and Technology, where he has been with since February 2003. He was a visiting research scholar in the Department of Electrical Engineering at the University of Houston from June to September 2013. He was the co-recipient of "The Fred Ellersick MILCOM Award for Best Paper in the Unclassified Technical Program" at the IEEE MILCOM'98, Boston, MA. His research interests lie in the areas of communication systems, statistical signal processing, and channel coding techniques.

Yunghsiang S. Han (S'90-M'93-SM'08-F'11) was born in Taipei, Taiwan, 1962. He received B.Sc. and M.Sc. degrees in electrical engineering from the National Tsing Hua University, Hsinchu, Taiwan, in 1984 and 1986, respectively, and a Ph.D. degree from the School of Computer and Information Science, Syracuse University, Syracuse, NY, in 1993. He was from 1986 to 1988 a lecturer at Ming-Hsin Engineering College, Hsinchu, Taiwan. He was a teaching assistant from 1989 to 1992, and a research associate in the School of Computer and Information Science, Syracuse University from 1992 to 1993. He was, from 1993 to 1997, an Associate Professor in the Department of Electronic Engineering at Hua Fan College of Humanities and Technology, Taipei Hsien, Taiwan. He was with the Department of Computer Science and Information Engineering at National Chi Nan University, Nantou, Taiwan from 1997 to 2004. He was promoted to Professor in 1998. He was a visiting scholar in the Department of Electrical Engineering at University of Hawaii at Manoa, HI from June to October 2001, the SUPRIA visiting research scholar in the Department of Electrical Engineering and Computer Science and CASE center at Syracuse University, NY from September 2002 to January 2004 and July 2012 to June 2013, and the visiting scholar in the Department of Electrical and Computer Engineering at University of Texas at Austin, TX from August 2008 to June 2009. He was with the Graduate Institute of Communication Engineering at National Taipei University, Taipei, Taiwan from August 2004 to July 2010. From August 2010, he is with the Department of Electrical Engineering at National Taiwan University of Science and Technology as Chair professor. His research interests are in error-control coding, wireless networks, and security. Dr. Han was a winner of the 1994 Syracuse University Doctoral Prize and a Fellow of IEEE.One of his papers won the prestigious 2013 ACM CCS Test-of-Time Award in cybersecurity.

Mengistu Abera Mulatu received the B.Tech. degree in Electrical Engineering from Defence University College, Debre Zeit, Ethiopia in 2002, and the M.S. degree in Electrical Engineering from Addis Ababa University, Addis Ababa, Ethiopia in 2007. He is currently working toward the Ph.D. degree in electrical engineering at National Taiwan University of Science and Technology, Taipei, Taiwan. His research interests are in the areas of the cooperative communications and system optimization for green wireless communications.

Li-Chung Chang received the B.S. degree in electrical engineering from National Tsing Hua University, Hsinchu, Taiwan, in 1990, the M.S. degree in electrical engineering from the Massachusetts Institute of Technology, Cambridge, MA, in 1994, and the Ph.D. degree in electrical engineering from Purdue University, West Lafayette, IN, in 2003. Currently, he is an Assistant Professor with the Department of Electrical Engineering at National Taiwan University of Science and Technology, Taipei, Taiwan. His research interests include MIMO OFDM systems and energy efficient communications.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2014.2312727, IEEE Transactions on Vehicular Technology

APPENDIX

Equation (5) can be further simplified as follows:

$$\begin{aligned} & \Pr(\boldsymbol{y}|\hat{\boldsymbol{v}},\hat{\boldsymbol{s}})\Pr(\hat{\boldsymbol{v}})\Pr(\hat{\boldsymbol{s}}) \geq \Pr(\boldsymbol{y}|\tilde{\boldsymbol{v}},\tilde{\boldsymbol{s}})\Pr(\tilde{\boldsymbol{v}})\Pr(\tilde{\boldsymbol{s}}) \\ \Leftrightarrow \quad & \Pr(\boldsymbol{y}|\hat{\boldsymbol{v}},\hat{\boldsymbol{s}})\Pr(\hat{\boldsymbol{s}}) \geq \Pr(\boldsymbol{y}|\tilde{\boldsymbol{v}},\tilde{\boldsymbol{s}})\Pr(\tilde{\boldsymbol{s}}) \ . \end{aligned} \tag{23}$$

Decomposing $\Pr(\boldsymbol{y}_0^{j+1}|\tilde{\boldsymbol{v}}_0^{j+1},\tilde{\boldsymbol{s}}_0^{j+1})\Pr(\tilde{\boldsymbol{s}}_0^{j+1})$ yields the following:

$$\begin{aligned} & \Pr(\boldsymbol{y}_0^{j+1}|\tilde{\boldsymbol{v}}_0^{j+1},\tilde{\boldsymbol{s}}_0^{j+1})\Pr(\tilde{\boldsymbol{s}}_0^{j+1}) \\ = \ & \Pr(y_{j+1},\boldsymbol{y}_0^{j}|\tilde{\boldsymbol{v}}_0^{j+1},\tilde{\boldsymbol{s}}_0^{j+1})\Pr(\tilde{\boldsymbol{s}}_0^{j+1}) \\ = \ & \Pr(\boldsymbol{y}_0^{j}|\tilde{\boldsymbol{v}}_0^{j+1},\tilde{\boldsymbol{s}}_0^{j+1})\Pr(y_{j+1}|\boldsymbol{y}_0^{j},\tilde{\boldsymbol{v}}_0^{j+1},\tilde{\boldsymbol{s}}_0^{j+1})\Pr(\tilde{\boldsymbol{s}}_0^{j+1}) \\ = \ & \Pr(\boldsymbol{y}_0^{j}|\tilde{\boldsymbol{v}}_0^{j+1},\tilde{\boldsymbol{s}}_0^{j+1})\Pr(y_{j+1}|\tilde{v}_{j+1},\tilde{s}_{j+1})\Pr(\tilde{s}_{j+1}|\tilde{s}_j) \\ & \cdot \Pr(\tilde{\boldsymbol{s}}_0^{j}) \\ = \ & \Big(\Pr(\boldsymbol{y}_0^{j}|\tilde{\boldsymbol{v}}_0^{j},\tilde{\boldsymbol{s}}_0^{j})\Pr(\tilde{\boldsymbol{s}}_0^{j})\Big)\Pr(y_{j+1}|\tilde{v}_{j+1},\tilde{s}_{j+1}) \\ & \cdot \Pr(\tilde{s}_{j+1}|\tilde{s}_j) \ . \end{aligned} \tag{24}$$

Taking the logarithms on both sides of (23) and substituting (24) into (23) yields the following:

$$\begin{aligned} & \sum_{j=0}^{N-1} \ln\left[\Pr(y_j|\hat{v}_j,\hat{s}_j)\Pr(\hat{s}_j|\hat{s}_{j-1})\right] \\ \geq \ & \sum_{j=0}^{N-1} \ln\left[\Pr(y_j|\tilde{v}_j,\tilde{s}_j)\Pr(\tilde{s}_j|\tilde{s}_{j-1})\right] \ , \end{aligned} \tag{25}$$

where $\tilde{s}_{-1}$ is the initial channel state. Thus, the right side of (25) acts as the path metric of $(\tilde{v}_j,\tilde{s}_j)$ for the proposed decoding metric, and the term inside the summation is the bit metric of $(\tilde{v}_j,\tilde{s}_j)$ which is expressed as follows:

$$\beta(y_j|\tilde{v}_j,\tilde{s}_j,\tilde{s}_{j-1}) = \ln[\Pr(y_j|\tilde{v}_j,\tilde{s}_j)] + \ln[\Pr(\tilde{s}_j|\tilde{s}_{j-1})] \ . \tag{26}$$

By considering all possible transition probabilities for the corresponding channel states, (25) can be expanded as follows:

$$\begin{aligned} & \sum_{j=0}^{N-1}\Big\{(1-b_j)\ln\left[\Pr(y_j|\hat{v}_j,\hat{s}_j=G)\right] + (b_j)\ln\left[\Pr(y_j|\hat{v}_j,\hat{s}_j=B)\right]\Big\} \\ & \quad + \sum_{j=0}^{N-1}\{(1-b_{j-1})(1-b_j)\ln(\Pr(\hat{s}_j=G|\hat{s}_{j-1}=G)) \\ & \qquad + b_{j-1}(1-b_j)\ln(\Pr(\hat{s}_j=G|\hat{s}_{j-1}=B)) \\ & \qquad + (1-b_{j-1})b_j\ln(\Pr(\hat{s}_j=B|\hat{s}_{j-1}=G)) \\ & \qquad + b_{j-1}b_j\ln(\Pr(\hat{s}_j=B|\hat{s}_{j-1}=B))\} \\ \geq & \sum_{j=0}^{N-1}\Big\{(1-b_j)\ln\left[\Pr(y_j|\tilde{v}_j,\tilde{s}_j=G)\right] + (b_j)\ln\left[\Pr(y_j|\tilde{v}_j,\tilde{s}_j=B)\right]\Big\} \\ & \quad + \sum_{j=0}^{N-1}\{(1-b_{j-1})(1-b_j)\ln(\Pr(\tilde{s}_j=G|\tilde{s}_{j-1}=G)) \\ & \qquad + b_{j-1}(1-b_j)\ln(\Pr(\tilde{s}_j=G|\tilde{s}_{j-1}=B)) \\ & \qquad + (1-b_{j-1})b_j\ln(\Pr(\tilde{s}_j=B|\tilde{s}_{j-1}=G)) \\ & \qquad + b_{j-1}b_j\ln(\Pr(\tilde{s}_j=B|\tilde{s}_{j-1}=B))\} \ , \end{aligned} \tag{27}$$

where $b_j = 0$, indicating that the $(j+1)^{th}$ channel state is good ($\tilde{s}_j = G$), and $b_j = 1$ indicates that the $(j+1)^{th}$ channel state is bad ($\tilde{s}_j = B$).

The PDF of the received bit conditioned on the channel state and the codeword bit for both good and bad channel states in which the noise statistics are known to the decoder are expressed as follows:

$$\Pr(y_j|\tilde{v}_j,\tilde{s}_j=G) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(\frac{-(y_j-(-1)^{\tilde{v}_j}\sqrt{E_s})^2}{2\sigma^2}\right), \tag{28}$$

$$\Pr(y_j|\tilde{v}_j,\tilde{s}_j=B) = \frac{1}{\sqrt{2\pi R\sigma^2}}\exp\left(\frac{-(y_j-(-1)^{\tilde{v}_j}\sqrt{E_s})^2}{2R\sigma^2}\right) \ . \tag{29}$$

Consequently, substituting (28) and (29) into (27) yields (8).