On Optimal Erasure and List Decoding Schemes of Convolutional Codes

Eran Hof Igal Sason Shlomo Shamai
Department of Electrical Engineering
Technion – Israel Institute of Technology
Haifa 32000, Israel
E-mails: {hof@tx, sason@ee, sshlomo@ee}.technion.ac.il

***Abstract*—A modified Viterbi algorithm with erasures and list-decoding is introduced. This algorithm is shown to yield the optimal decoding rule of Forney with erasures and variable list-size. For the case of decoding with erasures, the optimal algorithm is compared to the simple algorithm of Yamamoto and Itoh. The comparison shows a remarkable similarity in simulated performance, but with a considerably reduced decoding complexity.**

I. Introduction

The Viterbi algorithm (VA) was originally introduced for decoding convolutional codes over memoryless channels (see [4], [15] and [16]). The algorithm is popular in many coding and signal processing applications due to two important reasons: the VA is practical, and it yields the maximum-likelihood (ML) solution. There are several important generalizations of the VA, both in coding theory and signal processing. In this paper, generalizations of the VA for two coding problems are studied: list decoding, and decoding with feedback.

In list decoding problems, the decoder is allowed to output several candidates for each transmitted message. This set is called the decoded list. The event where the transmitted codeword is not in the decoded list, is called a *list-decoding error*. Several generalizations of the VA for list decoding applications are reported in the literature (see, e.g., [2], [5] [10], [11], [12], [13], [14]).

In practice, the decoder at the receiver may use a feedback channel to communicate with the transmitter. An important particularization of this setting is the case where the receiver can request for a retransmission of a message. Such a scheme is called a hybrid automatic repeat request (ARQ) scheme. The decoder in such a scheme either decodes the received signal and outputs an estimate for the transmitted message, or else declares an erasure. The decoding erasure triggers the transmitter, via the feedback channel, to retransmit its message. Several adaptations of the VA to support hybrid ARQ schemes are reported in the literature (see, e.g., [6], [7], [9], [17]).

Maximum-likelihood (ML) decoding is optimal with respect to the block error probability. For the case of decoding with erasures, there are two error events to be considered: the first is the event where the decoder outputs an incorrect decision, and this event is called an *undetected error event*. The second event is an *erasure*, where the decoder does not make a decision based on the received signal. Intuitively, it is expected to reduce the undetected error probability when the decoder is allowed to increase the erasure rate. As a result, the performance trade-off in the case of decoding with erasures, is between the undetected error probability and the total error probability (which includes undetected errors and erasures). The optimal decoding rule, with respect to this performance trade-off, is derived by Forney [3]. For list decoding, the rule provided in [3] is optimal with respect to the trade-off between the average size of the list, and the list decoding error probability. Generally, the optimal decoding rule of Forney is prohibitively complex, similarly to ML decoding in general. Performance bounds for erasure, list and feedback schemes with linear block codes were recently derived in [8]. These bounds are applicable to random and structured code ensembles, and their use is exemplified in [8].

The VA is known to yield the ML estimate for a finite-state Markov process observed via memoryless channels. For the particular case of coded communications with convolutional codes, the output of the VA coincides with the ML decision. In this paper, decoding with erasures and list decoding are concerned. A practical modification of the VA is introduced, which coincides with the optimal decoding rule of Forney for the case at hand. Although presented for decoding convolutional codes, this result is valid for the more general case of finite-state Markov processes observed via memoryless channels.

The simulated performance of the proposed modification is compared in this paper with the simulated performance of two suboptimal decoding algorithm with erasures: the likelihood-ratio (LR) test decoding rule, and a simple decoding scheme with repeat requests provided by Yamamoto and Itoh in [17]. Even though the decoding scheme in [17] is remarkably simple, the comparison shows good similarity between the performance of the simple scheme to the optimal one. On the other hand, the performance of the decoding algorithm based on the LR test is considerable degraded in comparison with that of the optimal performance.

This paper is structured as follows: Section II provides a short background, Section III proposes a modification to the VA, and Section IV presents some numerical results for the optimal decoding of convolutional codes with erasures.

II. A Preliminary on Decoding with Erasures and a Variable List-Size

Let $\mathcal{C} = \{\mathbf{x}_m\}_{m=1}^M$ be a codebook of size M and block length N, and consider a decoding rule that is characterized by a set of decoding regions $\{\Lambda_m\}_{m=1}^M$. That is, if the received vector $\mathbf{y}$ is in Λ_m, then the decoded codeword is $\mathbf{x}_m$. Under ML decoding, all the decoding regions are disjoint, and their union covers the entire observation space of received vectors. However, for the considered schemes where the decoder is allowed to declare on erasures and to output lists with a variable list-size, the decision regions $\{\Lambda_m\}_{m=1}^M$ are not necessarily disjoint, nor they include all possible received vectors. In the event where a received vector does not belong to a possible decoding region, the decoder declares an erasure, and the event is called an erasure event. The event where the decoder outputs an incorrect message, is called an undetected error event.

If the decoding regions are not disjoint, it is possible that a received vector belongs to several decision regions. In that case, the decoder outputs a list of all possible codewords (hence the term list decoding). The event where the transmitted codeword is not included in the list, is called a list decoding error. Note that the presented setting allows the decoding list to be with a variable size. In the following, such decoding rules are called *generalized decoding rules*.

Decoding with erasures and list-decoding with a variable list size, are first introduced in [3]. The optimal decoding rule with respect to the trade-off between the block error event (which includes both erasures and undetected errors) and the undetected error event is derived in [3]. Specifically, it is guaranteed that no other decoding rule exists for which lower error and undetected error probabilities are simultaneously obtained.

Definition 1 (**Forney's generalized decoding).** Consider a block code $\{\mathbf{x}_m\}_{m=1}^M$ over an alphabet $\mathcal{X}$. The generalized decoding rule is defined by the following decision regions:

$$\Lambda_m = \left\{ \mathbf{y} \in \mathcal{Y}^N : \ \frac{\Pr(\mathbf{y}, \mathbf{x}_m)}{\sum_{m' \neq m} \Pr(\mathbf{y}, \mathbf{x}_{m'})} \geq e^{NT} \right\} \quad (1)$$

where m is the codeword index, $T \in \mathbb{R}$ is a parameter, $\Pr(\mathbf{y}, \mathbf{x}_m)$ denotes the joint probability that $\mathbf{x}_m$ is the transmitted codeword and $\mathbf{y}$ is the received vector, and the summation is taken over all the codewords except for $\mathbf{x}_m$.

Remark 1 (**On list decoding).** The decoding rule in (1) is optimal for the case of list decoding. Specifically, no other decoding rule exists for which a lower list error probability and a smaller average size of the decoding list are simultaneously obtained.

Remark 2. The threshold parameter T in (1) controls the trade-off between erasures and undetected errors (or average list size and list decoding errors). Setting $T > 0$ guarantees the decision regions $\{\Lambda_m\}_{m=1}^M$ to be disjoint.

Remark 3. The decision region in (1) can be expressed equivalently in the form

$$\Lambda_m = \left\{ \mathbf{y} \in \mathcal{Y}^N : \ \Pr(\mathbf{x}_m | \mathbf{y}) \geq \frac{e^{NT}}{1 + e^{NT}} \right\}. \quad (2)$$

Consequently, if a codeword is selected according to the decoder with the decision regions in (2) with $T = 0$, then the same decision is made by a MAP decoder (as no other codeword can get an a-posteriori probability larger than $\frac{1}{2}$).

Remark 4 (**A likelihood-ratio (LR) decoding rule).** A suboptimal Likelihood-ratio (LR) decoding rule is also suggested in [3] where the summation in the denominator of (1) is replaced with the summand which corresponds to the second most probable codeword. The resulting decision regions are given by

$$\Lambda_m^{\text{LR}} = \left\{ \mathbf{y} \in \mathcal{Y}^N : \ \frac{\Pr(\mathbf{y}, \mathbf{x}_m)}{\Pr(\mathbf{y}, \mathbf{x}_{m_2})} \geq e^{NT} \right\} \quad (3)$$

where $m_2 = m_2(\mathbf{y})$ denotes the second most probable codeword for each received vector $\mathbf{y}$. Setting $T = 0$ in (3), the LR decoding rule coincides with ML decoding; note however that this observation does not follow for the optimal decoding rule in (1).

III. Optimal Generalized Decoding of Convolutional Codes over Memoryless Channels

In this section, a modified VA is presented for optimal decoding of convolutional codes with erasures. In addition, it is proved that this modification coincides with the optimal decoding rule in (1).

Assuming that all codewords are transmitted with equal a-priori probability, the joint probabilities in (1) can be replaced with conditional probabilities, and the decoding regions in (1) are given by:

$$\Lambda_m = \left\{ \mathbf{y} \in \mathcal{Y}^N : \ \frac{\Pr(\mathbf{y} | \mathbf{x}_m)}{\sum_{m' \neq m} \Pr(\mathbf{y} | \mathbf{x}_{m'})} \geq e^{NT} \right\}. \quad (4)$$

The standard VA provides the ML decision and its corresponding likelihood metric for the case at hand. Consequently, it remains to evaluate the denominator in (4) which is involved in the specification of the decision regions in [3].

Remark 5. Since

$$\frac{\Pr(\mathbf{y}, \mathbf{x}_m)}{\sum_{m' \neq m} \Pr(\mathbf{y}, \mathbf{x}_{m'})} = \frac{\Pr(\mathbf{y}, \mathbf{x}_m)}{\Pr(\mathbf{y}) - \Pr(\mathbf{y}, \mathbf{x}_m)}$$

the denominator of the LHS of the inequality in (1) can also be evaluated using the forward part of the Bahl, Cocke, Jelinek, and Raviv (BCJR) algorithm [1].

A convolutional code $\mathcal{C}$ with k inputs and n outputs for every time unit, and of memory length m is considered. The information sequence $\mathbf{u} = (\mathbf{u}_1, \ldots \mathbf{u}_B)$, of length kB symbols, is encoded (followed by a termination sequence) to form the codeword $\mathbf{x} = (\mathbf{x}_1, \ldots \mathbf{x}_{B+m})$ of length $n(B+m)$ symbols. We assume a memoryless channel, and denote the received sequence by $\mathbf{y}$. Each encoding operation, where k new inputs are introduced and n coded symbol outputs are transmitted at every time unit, is considered as a single time step. Let the metric for each branch in the trellis graph of $\mathcal{C}$ be

$$\mu(\mathbf{y}_t|\mathbf{x}_t) \triangleq \ln\big(p(\mathbf{y}_t|\mathbf{x}_t)\big), \quad 1 \leq t \leq B+m \qquad (5)$$

where $\mathbf{y}_t$ is the vector of n received samples at the decoder for the time step t, and $\mathbf{x}_t$ is the vector of coded symbols which corresponds to the considered branch at time t. In addition, we define the (cumulative) metric for each path in the trellis of $\mathcal{C}$ by

$$\mu(\mathbf{y}^t|\mathbf{x}^t) \triangleq \sum_{i=1}^{t} \mu(\mathbf{y}_i|\mathbf{x}_i)$$

where $\mathbf{y}^t = (\mathbf{y}_1, \ldots, \mathbf{y}_t)$ is the vector of nt received samples up to time step t, $\mathbf{x}^t = (\mathbf{x}_1, \ldots, \mathbf{x}_t)$ is the set of nt coded symbols of the considered path, and the sum is taken over all the t branches of this path. The set of nodes at a given time step t, which correspond to the possible encoder states in this time step, is denoted by $\mathcal{V}(t)$. For each node v in the trellis, the set of branches entering v is denoted by $\mathcal{B}_v$. The originating node of a trellis branch b, is denoted by v_b^{-1}, and the vector of output coded symbols of b is denoted by $\mathbf{x}(b)$.

A detailed description of the proposed algorithm is provided in Fig. 1. For the sake of simplicity, the algorithm in Fig. 1 is provided for the particular case of decoding a terminated convolutional code with erasures ($T > 0$). Steps 1(a) and (b) in Fig. 1 form the initialization actions for the standard VA. Starting from time step $t = m$, there exists a single surviving path in the trellis for each state whose cumulative metric is updated and stored. While proceeding along the trellis, steps 2(a)-2(d) in Fig. 1 are the familiar add-compare-select steps of the standard VA; for each state, the surviving path metric is chosen according to the maximal accumulate metric. Steps 1(c), 2(e) and 2(f) in Fig. 1 are the introduced modification. These steps allow the recursive evaluation of the sum in the denominator of (4). After this recursive evaluation along the trellis, the surviving path is selected, and the information bits are reconstructed according to the generalized decision rule in (4); else, an erasure is declared.

The following theorem assures that the suggested algorithm coincides with Forney's generalized decoding rule, as given in Definition 1:

Theorem 1. Consider the decoding of a terminated convolutional code using the algorithm in Fig. 1. Assuming that $\mathbf{x}_m$ is the codeword which corresponds to the surviving path, then the generalized metric μ_{G} satisfies:

$$e^{\mu_{\mathrm{G}}} = \sum_{m' \neq m} \Pr(\mathbf{y}|\mathbf{x}_{m'}).$$

Proof: Let $\mathcal{K}(v)$ denote the set of all possible paths entering a node v in the trellis graph of $\mathcal{C}$, except for the surviving path for v. We prove by induction that the generalized metric $\mu_{\mathrm{G}}(v)$ evaluated at $v \in \mathcal{V}(t)$ satisfies

$$e^{\mu_{\mathrm{G}}(v)} = \sum_{k \in \mathcal{K}(v)} \Pr(\mathbf{y}^t|\mathbf{x}_k^t) \qquad (6)$$

where $\mathbf{y}^t$ is the received vector up to time t (included), and $\mathbf{x}_k^t$ is the vector of the first nt symbols of the k-th codeword. First, we check that (6) follows for $t = m$ where each state $v \in \mathcal{V}(m)$ has a single entering path. Hence, the sum in (6) is void (i.e., $\mathcal{K}(v) = \emptyset$) which coincides with the setting $\mu_{\mathrm{G}}(v) = -\infty$ for all $v \in \mathcal{V}(m)$ (step 1(c) in Fig. 1). Assume by induction that (6) holds for $t = \tau - 1 \geq m$, and it is required to prove that (6) also holds for the next time step $t = \tau$. Let $\mathcal{K}_{\mathrm{s}}(v) \subseteq \mathcal{K}(v)$ denote all the paths in $\mathcal{K}(v)$ entering v via the same branch as the survivor. For $t = \tau$, consider the temporary result after step 2(e) in Fig 1, and assume that the algorithm is currently handling the state $v \in \mathcal{V}(\tau)$. Following the induction assumption, the temporary value of the generalized metric $\mu_{\mathrm{G}}(v)$ satisfies

$$\begin{aligned} e^{\mu_{\mathrm{G}}(v)} &= e^{\tilde{\mu}_{b^*}} \cdot e^{\mu_{\mathrm{G}}(v_{b^*}^{-1})} \\ &\overset{\text{(a)}}{=} e^{\tilde{\mu}_{b^*}} \sum_{k \in \mathcal{K}(v_{b^*}^{-1})} \Pr(\mathbf{y}^{\tau-1}|\mathbf{x}_k^{\tau-1}) \\ &\overset{\text{(b)}}{=} \sum_{k \in \mathcal{K}_{\mathrm{s}}(v)} \Pr(\mathbf{y}^{\tau}|\mathbf{x}_k^{\tau}) \end{aligned} \qquad (7)$$

where $\mathbf{y}^t$ is the received vector up to time step t, $\mathbf{x}_k^t$ is vector of the first nt symbols of the codeword corresponding to a path k in the trellis graph, b^* is the entering branch of the maximizing path metric in step 2(b) of the algorithm, $v_{b^*}^{-1}$ is the source node of b^*. Equality (a) follows by the induction assumption for $t = \tau - 1$, and equality (b) follows from the memoryless property of the channel and the definition of the branch metric in (5).

Next, let b be a branch which is handled by the algorithm in step 2(f.i), and denote by $\mathcal{K}_b(v) \subseteq \mathcal{K}(v)$ the set of all the paths in $\mathcal{K}(v)$ entering v via the branch b. After step 2(f.i) terminates, the variable ζ satisfies:

$$\begin{aligned} e^{\zeta} &\overset{\text{(a)}}{=} e^{\mu_{\mathrm{G}}(v_b^{-1}) + \tilde{\mu}_b} + e^{\mu(v_b^{-1}) + \tilde{\mu}_b} \\ &\overset{\text{(b)}}{=} \sum_{k \in \mathcal{K}_b(v)} \Pr(\mathbf{y}^{\tau}|\mathbf{x}_k^{\tau}) \end{aligned} \qquad (8)$$

1) For each state $v \in \mathcal{V}(m)$:
   a) Set the single path entering v as the survivor $\mathbf{s}(v)$.
   b) Evaluate the surviving entering path metric $\mu(v)$.
   c) Set the generalized metric $\mu_{\mathrm{G}}(v) = -\infty$.
2) Iterations over $m+1 \leq t \leq L+m$. For each state $v \in \mathcal{V}(t)$ do:
   a) Evaluate for each entering branch $b \in \mathbf{B}(v)$:
   $$\mu_b = \tilde{\mu}_b + \mu(v_b^{-1})$$
   where $\tilde{\mu}_b$ is the branch metric of b, and $\mu(v_b^{-1})$ is the path metric of the survivor at the source node v_b^{-1} of b.
   b) Find the entering branch with the maximal path metric:
   $$b^* = \arg\max_{b \in \mathbf{B}(v)} \mu_b.$$
   c) Set an updated survivor: $\mathbf{s}(v) = \big(\mathbf{s}(v_{b^*}^{-1}), \mathbf{x}(b^*)\big)$.
   d) Set an updated survivor path metric: $\mu(v) = \mu_{b^*}$.
   e) Evaluate a temporary generalized metric:
   $$\mu_{\mathrm{G}}(v) = \mu_{\mathrm{G}}(v_{b^*}^{-1}) + \tilde{\mu}_{b^*}$$
   where $\mu_{\mathrm{G}}(v_{b^*}^{-1})$ is the generalized metric evaluated at the previous node $v_{b^*}^{-1}$ of the survivor path, and $\tilde{\mu}_{b^*}$ is the branch metric of the last branch of the survivor path.
   f) For each of the rest of the entering branches $b \in \mathbf{B}(v) \setminus \{b^*\}$ do:
      i) Evaluate
      $$\zeta \triangleq \tilde{\mu}_{\mathrm{b}} + \max\big(\mu_G(v_b^{-1}), \mu(v_b^{-1})\big) + \ln\Big(1 + \exp\big(-\left|\mu_G(v_b^{-1}) - \mu(v_b^{-1})\right|\big)\Big)$$
      where $\tilde{\mu}_b$ is the branch metric of b, $\mu_G(v_b^{-1})$ and $\mu(v_b^{-1})$ are the generalized and standard path metrics at the initial node v_b^{-1} of the branch b.
      ii) Update the generalized metric:
      $$\mu_{\mathrm{G}}(v) = \max\big(\mu_G(v), \zeta\big) + \ln\Big(1 + \exp\big(-\left|\mu_G(v) - \zeta\right|\big)\Big).$$
3) If $\mu - \mu_{\mathrm{G}} > n(B+m)T$, return the survivor in the single node in $\mathcal{V}(B+m)$.
   Else, return an erasure.

Fig. 1: Modified VA for optimal generalized decoding (with erasures) of terminated convolutional codes.

where $\mathbf{y}^\tau$ forms the received vector up to time step τ, and $\mathbf{x}_k^\tau$ forms the sequence of the first $n\tau$ symbols of the codeword corresponding to a path k in the trellis, equality (a) follows from the equality

$$\ln(e^a + e^b) = \max(a, b) + \ln\left(1 + e^{-|a-b|}\right) \quad (9)$$

and equality (b) follows from the induction assumption, using the same arguments leading to (7). Finally, from (7)-(9), the update in step 2(f.ii) guarantees that (6) follows for $t = \tau$. Hence, by induction (6) follows for all $t \geq m$. ∎

Remark 6. The complexity of the proposed algorithm is linear in the block length B, and is exponential in the constraint length m of the code. This is the same complexity characteristics as in the case of the standard VA.

Remark 7 (**On generalized decoding with variable–size list).** Consider the problem of generalized decoding with a variable list-size according to the optimal decoding rule in (1) (with $T < 0$). According to the random coding analysis in [3] for low rates, the decoded list size is small (it typically includes one codeword); however, for higher code rates, the decoded list is likely to increase exponentially with the block length. Consequently, when practical decoding is of interest, some fixed limit on the decoded list is set. The following two options are suggested: the first, is to apply the (parallel) list VA as in [13] with the evaluation of the generalized metric as applied in the algorithm stated in Fig. 1. At the final step, only the survivors satisfying the condition in step 3, are left in the decoded list. As long as the size of the decoded list of the optimal decoding rule in Definition 1 is below the predetermined size-limit, Theorem 1 assures that the suggested modification coincides with the optimal decoding rule in [3]. The second option is to apply the evaluation of the generalized metrics in a serial implementation of the list VA (see, e.g., [10], [13], [14]). The serial implementation of the list VA iteratively produces a sequence of probable codewords where each iteration produces the next most probable path in the trellis graph of the code. After each iteration, the generalized metric of the decoded path is checked to satisfy the condition in step 3 of the algorithm, and the iterations stop if the condition fails. This scheme iteratively produces the list of codewords according to the optimal decoding rule in Definition 1. Since an exponentially amount of iterations is not practical, the algorithm needs to be stopped after a predetermined upper limit on the number of possible iterations. The resulting decoded list equals to the list under the optimal decoding rule only if the size of the optimal list is not larger than the predetermined limit.

Remark 8 (**On knowledge of channel state information).** Let $\mathbf{x}$ and $\mathbf{y}$ be vectors of size N over the channel input and output alphabets, respectively. The path metric

$$\mu(\mathbf{y}|\mathbf{x}) \triangleq \ln\big(p(\mathbf{y}|\mathbf{x})\big).$$

may be replaced with an erroneous metric μ' which does not rely on the complete channel state information. Consequently, for some applications, the implementation of the VA does not require complete channel state information at the receiver. Take for example a BSC with a crossover probability p. For this case:

$$\mu(\mathbf{y}|\mathbf{x}) = d_{\mathrm{H}}(\mathbf{y},\mathbf{x}) \ln\left(\frac{p}{1-p}\right) + N\ln(1-p)$$

where $d_{\mathrm{H}}(\mathbf{y},\mathbf{x})$ is the Hamming distance between $\mathbf{x}$ and $\mathbf{y}$. Another example is the AWGN channel with energy E_{s} per transmitted symbol and a two-sided density noise power spectrum $N_0/2$, where we have:

$$\mu(\mathbf{y}|\mathbf{x}) = \frac{2E_{\mathrm{s}}}{N_0}\mathbf{y}^{\mathrm{T}}\mathbf{x} - \left(\frac{E_{\mathrm{s}}}{N_0}\left(\mathbf{y}^{\mathrm{T}}\mathbf{y}+N\right) - \frac{N}{2}\ln\frac{E_{\mathrm{s}}}{\pi N_0}\right).$$

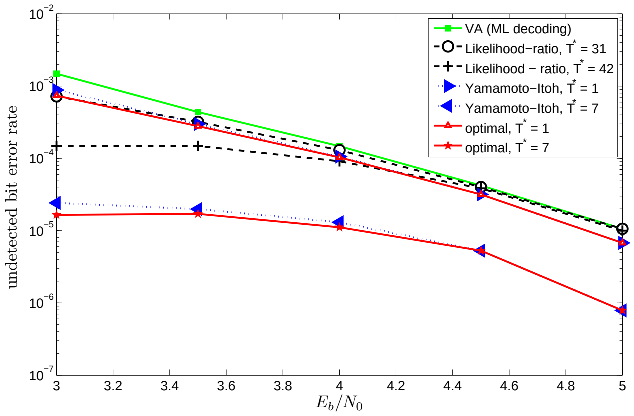
A weakness of the proposed algorithm is that its proof of optimality according to Theorem 1 does not necessarily follow if the metric μ is replaced with an erroneous metric. The implication of this observation is that the proposed algorithm may require complete channel state information to guarantee its optimality according to Theorem 1.

IV. Examples

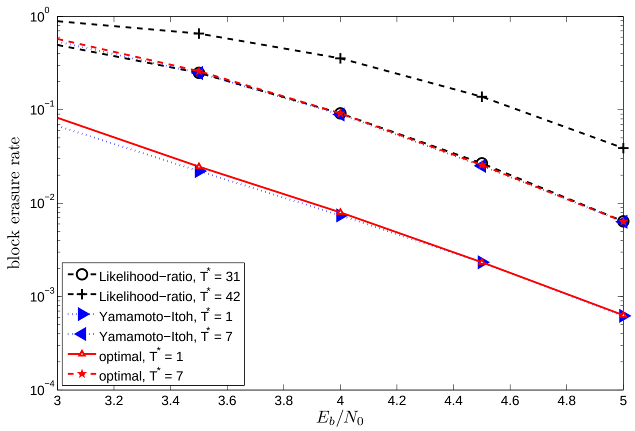
The performance of the (2,1,4) convolutional code with generator polynomials $g_1(D) = 1 + D + D^3 + D^4$ and $g_2(D) = 1 + D^3 + D^4$ is simulated under some generalized decoding algorithms with erasures. This code is used in the GSM Phase 2 system for the full-rate data traffic channel [18]. The results are provided for information sequence of 240 bits, with additional 4 bits of termination sequence (these bits are called “tail bits” in [18] where the same parameters are used). It is assumed that the transmission takes place over an AWGN channel with a binary phase shift keying (BPSK) modulation. Exact likelihood metrics are used in this simulation assuming complete channel state information at the decoder, i.e., the metric used in the simulation is $\mu(\mathbf{y}|\mathbf{x}) = 2\frac{E_s}{N_0}\mathbf{y}^{\mathrm{T}}\mathbf{x}$. Denote $T^* \triangleq n(B+m)T$, then the threshold e^{NT} in (1) and (3), is equal to e^{T^*}. In the following simulated results, the error performance for different values of the threshold parameter T^* are plotted. Note however, that when T^* is fixed with the applied metric, it follows that a different receiver is simulated for each SNR value. In Fig. 2(a), the undetected bit error rates under the optimal generalized decoding algorithm in Fig. 1 (based on the optimal decoding rule of Forney [3]), with $T^* = 1$ and 7, are provided. In addition, the bit error rate of the standard VA and the undetected bit error rate of the LR decoding in (3) with $T^* = 31$ and 42, and under the decoding algorithm of Yamamoto-Itho (this algorithm uses a threshold A (see [17, Section II]), the same T^* values of the optimal algorithm are used for A.) (YI) [17] , are provided for comparison. The corresponding block erasure rates for the simulated algorithms are provided in Fig. 2(b). It is evident that the estimated performance of the optimal algorithm outperforms the one of the LR decoding rule. The undetected error performance of the optimal algorithm with $T^* = 1$, resembles the undetected bit error rates of the LR decoding rule with $T^* = 31$ and 42. However, the corresponding erasure rates under optimal decoding clearly outperform the suboptimal erasure rates under the LR decoding rule. Moreover, the optimal algorithm with $T^* = 7$, which results in similar erasure rates as the LR decoding rule with $T^* = 31$, whereas its undetected bit error rates clearly outperforms the undetected bit error rates under the LR decoding rule. Comparing the simulated performance of the optimal algorithm with the YI decoding algorithm shows a remarkable improvement as compared to the LR decoding rule, and a good match with the simulated performance under the optimal decoding rule. For high SNR values, both decoding algorithms show the almost the same performance. For low SNR values, the gain of the optimal algorithm as compared with the YI algorithm is marginal. Take for example the results for $T^* = 7$ where both decoding algorithms have almost the same erasure rates, while only a slight improvement of the undetected bit error rate is observed for the optimal algorithm (in low SNR values).

V. Summary and Conclusions

An optimal algorithm is provided based on the generalized decision regions of Forney [3]. This algorithm allows for a practical generalized decoding of convolutional codes with erasures and variable list-sizes. The simulated performance of the proposed algorithm is



(a)



(b)

Fig. 2: Error performance of a (2,1,4) convolutional code under generalized decoding with erasures. Undetected bit error rates, and erasure rates, are provided in plots (a) and (b), respectively, under the optimal decoding in Fig. 1, the LR decoding rule in (3), and for the Yamamoto-Itoh (YI) decoding algorithm [17]. The bit error rate under ML decoding (using the standard VA) is also provided. The results are provided for information sequence of 240 bits, with additional 4 bits of termination sequence

compared with two suboptimal erasure decoding algorithms: the LR decoding rule in (3), and an algorithm by Yamamoto and Itoh (YI) [17]. The difference between the simulated performance of the optimal decoding algorithm and the YI algorithm is negligible. Moreover, the implementation of the YI algorithm is simpler and it yields a remarkable reduction in decoding complexity. The performance of the LR decoding rule, on the other hand, are substantially inferior.

VI. Acknowledgement

This research was supported by the Israel Science Foundation (grant no. 1070/07), and by the European Commission in the framework of the FP7 Network of Excellence in Wireless Communications (NEWCOM++).

References

[1] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, “Optimal decoding of linear codes for minimizing symbol error rate,” *IEEE Trans. Information Theory*, vol. 20, no. 2, pp. 284–287, March 1974.

[2] B. Checn, and C.-E. W. Sundberg, “List Viterbi algorithm for continuous transmission,” *IEEE Trans. Communications*, vol. 49, no. 5, pp. 784–792, May 2001.

[3] G. D. Forney, “Exponenttial error bounds for erasure, list, and decision feedback schemes,” *IEEE Trans. Information Theory*, vol. 14, no. 2, pp. 206–220, March 1968.

[4] G. D. Forney, “The Viterbi algorithm,” *Proceedings of the IEEE*, vol. 61, no. 3, pp. 268-278, March 1973.

[5] T. Hashimoto, “A list-type reduced-constraint generalization of the Viterbi algorithm,” *IEEE Trans. Information Theory*, vol. 33, no. 6, pp. 866–876, September 1987.

[6] T. Hashimoto, “A coded ARQ scheme with the generalized Viterbi algorithm,” *IEEE Trans. Information Theory*, vol. 39, no. 2, pp. 423–432, March 1993.

[7] T. Hashimoto, “On the error exponent of convolutionally coded ARQ,” *IEEE Trans. Information Theory*, vol. 40, no. 2, pp. 567–575, March 1994.

[8] E. Hof, I. Sason and S. Shamai, “Performance bounds for erasure, list and feedback schemes with linear block codes,” submitted to the *IEEE Trans. on Information Theory*, May 2009. [Online]. Available: http://www.ee.technion.ac.il/people/sason/.

[9] B. D. Kudryashov, “Error probability for repeat request systems with convolutional codes,” *IEEE Trans. Information Theory*, vol. 39, no. 5, pp. 1680–1684, March 1993.

[10] L. Lijofi, D. Cooper, and B. Canpolat “A reduced complexity list single-wrong-turn (swt) Viterbi decoding algorithm,” *Proc. of the 15th IEEE International Symposion on Personal, Indoor and Mobile Radio Communications (PIMRC 2004)*, pp. 274–279, September 2004.

[11] T. Niinomi, T. Matsushima, and S. Hirasawa, “On the generalized Viterbi algorithm using likelihood ratio testing,” *Proc. of the 2002 IEEE International Symposion on Information Theory (ISIT 2002)*, Lausanne, Switzerland, Jun 30–July 5, 2002.

[12] M. Röder, and R. Hamzaoui, “Fast tree-trellis list Viterbi decoding,” *IEEE Trans. Communications*, vol. 54, no. 3, pp. 453–461, March 2006.

[13] N. Seshardi, and C.-E. W. Sundberg, “List Viterbi algorithms with applications,” *IEEE Trans. Communications*, vol. 42, no. 2/3/4, pp. 313–323, February/March/April 1994.

[14] N. Seshardi, and C.-E. W. Sundberg, “List and soft symbol output Viterbi algorithms: Extensions and comparisons,” *IEEE Trans. Communications*, vol. 43, no. 2/3/4, pp. 277–287, February/March/April 1995.

[15] A. J. Viterbi, “Error bounds for convolutional codes and an asymptotically optimum decoding algorithm,” *IEEE Trans. Information Theory*, vol. 13, no. 2, pp. 260–269, April 1967.

[16] A. J. Viterbi and J. K. Omura, *Principles of Digital Communication and Coding*, McGraw-Hill, 1979.

[17] H. Yamamoto and K. Itoh, “Viterbi decoding algorithm for convolutional codes with repeat request,” *IEEE Trans. Information Theory*, vol. 26, no. 5, pp. 540–547, March 1980.

[18] ETSI EN 300 909, “Digital cellular telecommunications system (Phase 2+); Channel coding (GSM 05.03),” *European Telecommunications Standards Institute 2000*, version 8.5.1, 1999.