Sequential Decoding of Convolutional Codes for Synchronization Errors

Anisha Banerjee, Andreas Lenz and Antonia Wachtet-Zeh
Institute for Communications Engineering
Technical University of Munich
DE-80333 Munich, Germany
Email: anisha.banerjee@tum.de, andreas.lenz@mytum.de, antonia.wachter-zeh@tum.de

Abstract—In this work, a sequential decoder for convolutional codes over channels that are vulnerable to insertion, deletion, and substitution errors, is described and analyzed. The decoder expands the code trellis by introducing a new channel state variable, called drift state, as proposed by Davey-MacKay. A suitable decoding metric on that trellis for sequential decoding is derived, in a manner that generalizes the original Fano metric. Under low-noise environments, this approach reduces the decoding complexity by a couple orders of magnitude in comparison to Viterbi’s algorithm. An analytical method to determine the computational cutoff rate is also suggested. This analysis is supported with numerical evaluations of bit error rates and computational complexity, which are compared with respect to optimal Viterbi decoding.

I. INTRODUCTION

Most error-control systems usually operate under the assumption of perfect synchronization between transmitter and receiver, while their respective decoders are designed to detect and correct substitution errors alone. However, this assumption might not hold in the case of some networking channels, and thus some transmitted symbols may be lost or random ones may be inserted into the received stream. Similar errors occur in DNA-based storage systems [1, 2], due to imperfections in the synthesis and sequencing processes. Such errors are referred to as deletions and insertions.

There exists rich literature dedicated to the study of channels that are susceptible to insertion, deletion and substitution errors, and suitable error-correcting codes to increase transmission reliability under such environments [3–10]. In this work, we are interested in the use of convolutional codes for the purpose of correcting these errors. Prior work in [11, 12] suggested new trellis structures that helped in adapting the conventional Viterbi and MAP decoders to handle insertions and deletions. Concatenated schemes [13] with inner convolutional codes have built on these trellis structures to decode from multiple sequences over insertion and deletion channels. One drawback of these decoding approaches however, lies in their memory requirements and computational complexity. In particular, the number of states in the decoding trellis grows rapidly with factors like code memory, number of information blocks per codeword and maximum allowable insertions and deletions per block. This motivated us to look into an alternative approach, namely sequential decoding.

First proposed by Wozencraft [14], sequential decoders constitute a decoding strategy that is suited to convolutional codes with higher constraint lengths. This is primarily owed to the fact that a typical sequential decoder will only examine those codewords that seem likely to have been transmitted, unlike the Viterbi decoder which assesses all possibilities, regardless of noise levels. Though this leads to a worse error-correcting performance, the resultant decoding complexity is reduced and effectively independent of the encoder’s memory.

The main objective of this work is to tailor the sequential decoding approach for use in channels that experience insertions, deletions as well as substitution errors. This problem was first addressed by Gallager [9], who used Wozencraft’s original algorithm to implement the sequential decoder, and subsequently analyzed its complexity. Mansour and Tewfik [15] also worked on this problem, by adopting a new trellis structure and specifically limiting their focus to the stack algorithm. However, unlike [9, 15], this work formulates a new decoding metric, wherein the likelihood component is computed using the lattice metric [16], and an additional bias term accounts for probability of the predicted message sequence and that of the received bit sequence. Furthermore, we employ the trellis structure proposed in [12] and limit our attention to Fano’s algorithm, which typically performs fewer computations and explores more paths compared to other variants of sequential decoding. Using the approach in [17], an analytical assessment of the algorithm’s average complexity is also performed.

II. PRELIMINARIES

A. Channel model

As in [5, 16], we adopt a finite state machine model for our channel, specified by three parameter $P_i$, $P_d$ and $P_s$, which denote the insertion, deletion and substitution probabilities of the channel, respectively. Let $x_i \in \{0, 1\}^M$ denote a sequence of bits awaiting transmission. From Fig. 1, we observe that under this construct, for each input bit, one of four events may occur: a random bit is inserted into the received stream with probability $P_i$ and $x_i$ remains in the transmission queue; or the next bit queued for transmission, i.e., $x_i$, is deleted with probability $P_d$; or $x_i$ is received at output end, either erroneously or correctly, with probabilities $P_s$ and $(1 - P_s)$ respectively. Here, $P_t = 1 - P_i - P_d$ simply refers to the transmission probability.

B. Convolutional codes

Before describing the decoding framework, we shortly recapitulate the basics of convolutional codes. These codes constitute a special category of tree codes, that incorporate memory and aim to encode a stream of input bits in a block-wise manner, by means of shift registers. They are typically specified by three parameters: $[c, b, m]$, indicating that for every $b$ input bits received, the encoder generates $c$ output bits, which are a function of the last $b(m + 1)$ input bits.
C. Joint code and channel tree structure

As in [12], the vector obtained at the receiving end of the channel is viewed as the output of a hidden Markov model (HMM), where each hidden state is a pair of the encoder state and the drift value. In this context, we define the drift [5] as the difference between number of bits received and transmitted. In particular, $d_i$ is used to signify the net drift accumulated after the transmission of $i$ bits. For more details about the drift variable, we refer the reader to [5].

The decoder works on a tree representation of this HMM, such that any path in this tree describes how the encoder state and net drift value could change over time. As a demonstration, Fig. 2 depicts the joint code and channel tree for a [3, 1, 1] convolutional encoder. For any given sequence of HMM states, the concatenation of edge labels along the respective path in the code tree indicates the originally transmitted codeword.

D. Fano’s algorithm

In this work, we limit our focus to a particular variant of sequential decoding, namely Fano’s algorithm [18]. It operates on the principle that node metrics along the correct path keep increasing on an average. The algorithm searches for such a path, by tracking metrics of the current, previous and best successor nodes, denoted as $\mu_{\text{path}}$, by tracking metrics of the current, previous and best successor nodes, denoted as $\mu_{\text{path}}$, respectively. A dynamic threshold $T$ is used to check for the aforementioned property. This variable can only be altered by integer multiples of a user-defined step size $\Delta$. Starting from tree root, the decoder works as follows:

1) If best successor has metric $\mu_s > T$, move forward.
   - If this node has never been visited before, $T$ is tightened such that
     \[ T \leq \mu_c < T + \Delta. \] (1)
   2) Else, step back to the immediate predecessor.
      - If other successors with metrics above $T$ exist, the decoder steps forward to it, as in step 1).
      - Else:
        - If $\mu_p < T$, lower $T$ by $\Delta$.
        - Else repeat step 2).

In this manner, all the paths with metrics above or equal to the threshold $T$ are systematically explored.

III. DECODER METRIC

From the preceding discussion, it is evident that a metric for each tree node must be defined to quantify its closeness to the received vector. Specifically, this metric should help minimize the probability of choosing a wrong successor.

Let $y_i^N = (y_1, \ldots, y_N)$ denote a received sequence produced by the transmission of a codeword of $N$ blocks. Now consider a path in the corresponding code tree for a convolutional code with parameters $[c, b]$, from the root to a node at depth $t$, say $v_0^t$, that corresponds to the sequence of convolutional code states $s_0^t = (s_0, \ldots, s_t)$ and the drift state vector $d_0^t = (d_0, d_c, \ldots, d_c)\text{1}$, where initial drift is $d_0 = 0$. Then, the metric of node $v_0^t$ in the tree is given by

\[ \mu(v_0^t) = \log_2 P(v_0^t | y_1^N) = \log_2 P(v_0^t, y_1^N) - \log_2 P(y_1^N). \] (2)

Hence, the decoder metric of a path is essentially the probability of its predicted codeword and drift changes, given a specific received frame. This definition is in the same spirit as that in [19], wherein Massey proved the optimality of the Fano metric in the context of binary symmetric channels.

Before further simplifying (2), we recognize that the path $v_0^t$ only accounts for the first $ct+d_c$ symbols of the received vector. The remaining symbols $y_{ct+d_c+1}^{ct+d_c+1}$, are assumed to have been produced by a tailing message sequence that guides the convolutional encoder through the states $s_{i+1}^L = (s_{i+1}, \ldots, s_L)$. Additionally assuming that $y_{ct+d_c+1}^N$ is unaffected by bits transmitted prior to it, we may write

\[ P(v_0^t, s_{i+1}^L, y_1^N) = P(v_0^t) P(s_{i+1}^L) P(y_1^N | v_0^t, s_{i+1}^L) \]
\[ = P(v_0^t, y_1^{ct+d_c}) P(s_{i+1}^L, y_1^{ct+d_c} | s_t, d_c). \]

Marginalizing this term over all possible message tails,

\[ P(v_0^t, y_1^N) = \sum_{s_{i+1}^L} P(v_0^t, s_{i+1}^L, y_1^N) \]
\[ = P(v_0^t, y_1^{ct+d_c}) \sum_{s_{i+1}^L} P(s_{i+1}^L, y_1^{ct+d_c} | s_t, d_c). \] (3)

1Since a node at depth $i$ is reached after the transmission of $ci$ bits.
2Exactly true for bits $y_{ct+d_c+1}^{ct+d_c+1}$.

Figure 1: Allowed transitions in the state machine model for the insertion, deletion, and substitution channel [5]

Figure 2: Joint code and channel tree of a [3, 1, 1] convolutional code. Dashed lines correspond to an input of 0 and solid lines to input 1. Each node has two state variables, the convolutional code state and the drift state.
Both equations (2) and (3) require us to evaluate the probability of receiving a particular sequence. To make the dependence of this quantity on the length of the causal transmitted sequence more explicit, we introduce the following notation.

\[ P_M(y^N) = \sum_{x \in \mathbb{F}_2^N} P(x, y^N). \]  \hspace{1cm} (4)

Applying (3) and (4) in (2), we arrive at the following definition of decoder metric.

\[
\mu(v_0^t) = \log_2 P(v_0^0, y_1^{c+d_*}) + \log_2 P_{c(L-t)}(y_{c+d_*+1}^{c+d_*+L})
- \log_2 P_{cL}(y_1^N) \\
= \sum_{i=0}^{t-1} \log_2 P(s_{i+1} | s_i) + \sum_{i=0}^{t-1} \log_2 P(d_{c(i+1)} | d_{ci})
+ \sum_{i=0}^{t-1} \log_2 P(y_{c(i+1)+d_{c(i+1)}} | s_{i+1}, d_{ci}, d_{c(i+1)})
+ \log_2 \frac{P_{c(L-t)}(y_{c+d_*+1}^{c+d_*+L} | d_{ci})}{P_{cL}(y_1^N)}. \]  \hspace{1cm} (5)

The final equality is due to the relative independence of consecutive message blocks and the Markov chain-like behavior of the sequence of drift values. In addition, since we only consider drift sequences with initial drift \(d_0 = 0\), we set \(P(d_0 = 0) = 1\). We thus define equivalent decoder metrics for individual branches of the code tree as

\[
Z(v_0^{t+1} | v_0^t) = \mu(v_0^{t+1}) - \mu(v_0^t)
= \log_2 P(s_{t+1} | s_t)
+ \log_2 P(y_{c(t+1)+d_{c(t+1)}} | s_{t+1}, d_{ct})
+ \log_2 \frac{P_{c(L-t)}(y_{c+d_*+1}^{c+d_*+L} | d_{ct})}{P_{cL}(y_1^N)}. \]  \hspace{1cm} (6)

A similar method can be used to evaluate the drift likelihood \(P(d_{c(t+1)} | d_{ct})\) by setting the horizontal, vertical and diagonal edge weights of the lattice to \(P_1, P_d\) and \(P_1\) respectively. Alternatively, one may use the closed-form expression specified in [8]. The quantity in (4) can also be evaluated similarly, by assuming that all transmitted sequences \(x \in \mathbb{F}_2^M\) are equally likely. In this case, the corresponding edge weights will simply change to \(\frac{P}{P_1}, \frac{P}{P_d}\) and \(\frac{P}{P_1}\) respectively.

To obtain an asymptotic expression for (6) as we will require in the next section, the methods from [20–22] are employed.

IV. COMPUTATIONAL CUT-OFF RATE

To assess the complexity of Fano’s decoder, it suffices to count the number of forward steps taken by the decoder, since each time a new node is visited, branch metrics for all immediate successors must be computed. This is clearly the most costly step in Fano’s algorithm. To establish an upper bound on the total number of visits to nodes in a given code tree, we adopt the approach and the modeling assumptions outlined in [17], which are summarized in the following.

This analysis hinges on the assumption of correct decoding. Consider a received vector \(y^N\), which results from the transmission of \(L\) blocks. Let the decoder output represent the path

\[ p^* = ((s_0^*, d_0^*), (s_1^*, d_1^*), \ldots (s_L^*, d_L^*)). \]

The remaining nodes that describe all the incorrect paths, are grouped into \(L\) subtrees: \(\tau_0, \ldots, \tau_{L-1}\). Here, \(\tau_i\) contains all nodes that hypothesize any false path that stems from \(v_i^0\), i.e., \(i\)’th node on the correct path, \(p^*\). If \(C(v_i^0)\) refers to the number of visits by decoder to node \(v_i^0\), then the total complexity of the complete decoding operation can be expressed as

\[
C_{\text{total}} = \sum_{i=0}^{L-1} \left( C(v_i^0) + \sum_{j=i+1}^{L-1} \sum_{v_j^0 \in \tau_i} C(v_j^0) \right). \]  \hspace{1cm} (7)

We average this quantity over a code ensemble with parameters \((c, b)\), and all possible transmitted and received sequences. For analytical simplicity, we confine our attention to codewords of infinite length. Doing so makes the code tree infinite in length, thereby allowing us to assume similar statistical properties for all \(\tau_i\). Consequently, the task of computational analysis reduces to determining the mean decoding complexity per block, which we denote by \(C_{\text{av}}\), i.e.,

\[
C_{\text{av}} = E[C(v_0^0)] = \frac{1}{L} E[C(v_0^0)]. \]  \hspace{1cm} (8)

The discussion in Section II-D suggests that a drop in metric along \(p^*\) causes the decoder to examine alternate paths in the incorrect subtrees. When none of these seem plausible, the decoder returns to previously visited nodes in \(p^*\), but with a lowered threshold, and can only move forward when the criterion \(\mu_s \geq \sigma\), is upheld. Thus, the number of visits to a node clearly depends on its metric.

Let \(\mu_{\text{min}}^c\) and \(T_{\text{min}}^c\) denote the minimum node metric and minimum threshold value along \(p^*\), during a complete decoding operation respectively. It is easy to see that a lower value of \(T_{\text{min}}^c\) spurs the decoder to track back more often to explore incorrect nodes in the tree. Hence, one may evaluate \(C_{\text{av}}\) as

\[
C_{\text{av}} = \sum_{i=0}^{\infty} P(T_{\text{min}}^c = -i\Delta) E[C(v_0^0) | T_{\text{min}}^c = -i\Delta]. \]  \hspace{1cm} (9)

The second equality follows from the fact that threshold always starts at 0, and only changes by the magnitude of \(\Delta\).

With an additional assumption that branch metrics along the correct path are independently and identically distributed, and that the same applies to branch metrics along incorrect paths for a specific drift change between adjacent nodes, we acquire a bound for \(C_{\text{av}}\), that unfortunately overshoots practical values by several orders of magnitude, akin to what the analysis in [23] for binary symmetric channels yields. However, it offers one key insight that \(C_{\text{av}}\) only converges if

\[
\sigma_0 + \sigma_1 < 0, \]  \hspace{1cm} (10)

where \(\sigma_0\) and \(\sigma_1\) are constants dependent on channel and code parameters. An intuitive interpretation of (10) is that for \(C_{\text{av}}\) to be finite, the rate of decline of probability \(P(\mu_{\text{min}}^c < y)\) by decreasing \(y\), i.e., \(\sigma_0\), should adequately compensate the rate of rise of computations in \(\tau_0\), i.e., \(\sigma_1\). When (10) holds with equality, the channel is said to operate at computational cutoff rate \(R_0\), which is the code rate beyond which using sequential decoding becomes computationally impractical. This is due to the fact that for rates exceeding \(R_0\), the severity of channel noise causes frequent backtracking and more computations in incorrect subtrees. For finite-length codewords, this implies that decoding complexity of a single frame grows linearly with
tree depth if (10) holds, and exponentially otherwise.

The upcoming derivations lead us to the bound on (9).

A. Distribution of minimum threshold

To bound the initial term in (9), we first evaluate the cumulative probability distribution of metric values at depth $i$ along $p^*$ as follows, for any $\sigma < 0$,

$$P(\mu^*_i \leq y) = P(2^{\sigma \mu^*_i} \geq 2^{\sigma y}) \leq 2^{-\sigma y} \mathbb{E}[2^{\sigma \mu^*_i}]$$

where $Z^*_i$ refers to the $i^{th}$ branch metric along $p^*$. Assuming all $Z^*_i$'s are independent and identically distributed yields

$$P(\mu^*_i \leq y) \leq 2^{-\sigma y} \mathbb{E}[2^{\sigma Z^*_i}] = 2^{-\sigma y} \mathbb{E}[2^{\sigma y}]$$

Here, $\mathbb{E}[2^{\sigma y}]$ is the moment generating function of branch metrics along the correct path. Choosing a proper $\sigma_0 < 0$ such that $\mathbb{E}[2^{\sigma y}] = 1$, we get $P(\mu^*_i \leq y) \leq 2^{-\sigma_0 y}$, leading to

$$P(\mu^*_\min \leq y) \leq 2^{-\sigma_0 y}$$

We can obtain a similar bound for $T^*_\min$ by noting that it can only be 0 or any negative multiple of $\lambda^*$, and applying (1).

$$P(T^*_\min \leq y) = P(T^*_\min \leq \frac{y}{\lambda^*} \Delta) = P(\mu^*_\min \leq \frac{y}{\lambda^*} \Delta + \Delta)$$

Using an indicator function $\phi(\nu^*_i)$ defined as

$$\phi(\nu^*_i) = \begin{cases} 1, & \text{if node } \nu^*_i \text{ is visited} \\ 0, & \text{else} \end{cases}$$

the previous inequality may be further refined to

$$C^*(\nu^*_i) \leq \left( \frac{\mu^*_i - T^*_\min}{\lambda^*} + 1 \right) \phi(\nu^*_i)$$

Since the metric of root $\nu^*_0$ is always set to zero, we may write

$$\mathbb{E}[C^*(\nu^*_0)|T^*_\min = y] \leq -\frac{y}{\Delta} + 1$$

$$+ \sum_{j=1}^{\infty} \mathbb{E} \left[ \sum_{\nu^*_j \in \tau_0} C^*(\nu^*_j)|T^*_\min = y \right]$$

The latter term is decomposed by subdividing $\tau_0$ into two subtrees: $\tau^*_0$, that contains nodes which hypothesize a wholly inaccurate message sequence, and $\tau^*_{\min}$, which consists of nodes that follow the true message sequence, at least initially, but suggest a different drift sequence.

Next, we characterize the distribution of node metrics, or equivalently branch metrics, in $\tau^*_0$. Given their dependence on relative drift changes, these branch metrics are treated in a drift-specific manner. Let $Z'(\delta)$ describe the branch distribution of metric values in $\tau^*_0$ for a specific drift change $\delta$. Also, let $i_{\max}$ and $d_{\max}$ denote the maximum allowable insertions and deletions over a single block. Since $\mu(\nu^*_i) = Z(\nu^*_0 \to \nu^*_i)$, the number of visits to nodes in $\tau^*_0$ at depth 1, is bounded by

$$\mathbb{E} \left[ \sum_{\nu^*_0 \in \tau^*_0} \left( \frac{Z(\nu^*_0 \to \nu^*_0) - y}{\Delta} + 1 \right) \phi(\nu^*_0)|T^*_\min = y \right]$$

$$= \left( 2^b - 1 \right) \sum_{\nu^*_0 \in \tau^*_0} \sum_{\nu^*_0 \to \nu^*_1 \leq \nu^*_0 \to \nu^*_1} \sum_{z_{1} \geq y} \left( \frac{z_{1} - y}{\Delta} + 1 \right) P(Z'(\delta) = z_1)$$

where $\lambda = i_{\max} + d_{\max} + 1$ and the variable $\zeta'$ uniformly combines the probability distributions of all the drift-specific random variables. By assuming that branch metrics across different levels are independently distributed, we can generalize (18) for higher depths.

$$\mathbb{E} \left[ \sum_{\nu^*_0 \in \tau^*_0} \left( \frac{\mu(\nu^*_0) - y}{\Delta} + 1 \right) \phi(\nu^*_0)|T^*_\min = y \right]$$

$$= \left( 2^b \right) \sum_{\nu^*_0 \in \tau^*_0} \sum_{\nu^*_0 \to \nu^*_1 \leq \nu^*_0 \to \nu^*_1} \sum_{\nu^*_0 \to \nu^*_1 \leq \nu^*_0 \to \nu^*_1} \sum_{z_{1} \geq y} \left( \frac{z_{1} - y}{\Delta} + 1 \right) f_{\nu^*_0}(y, \mu)$$

For instance, an all-zero codeword $\delta_{\nu^*_0} = \{0\}$, implies $\nu^*_0 \to \nu^*_1 \leq \nu^*_0 \to \nu^*_1$ and $z_{1} \geq y$. The resulting expression describes the distribution of node metrics in $\tau^*_0$. Evidently, branch metrics in $\tau^*_0$ follow a different probability distribution than $\tau^*_0$, since the predicted block output is not necessarily independent of the received frame, unlike the previous case. Hence, we let a random variable $Z^*(\delta)$ characterize the distribution of metrics along such branches, for a specific drift change $\delta$. We also note that multiple paths hypothesizing the same message sequence as $p^*$ but with alternate drift sequences, may yield the same node metrics as in $p^*$. For instance, an all-zero codeword or with $\nu^*_0 \to \nu^*_1 \leq \nu^*_0 \to \nu^*_1$ for all $\nu^*_0 \in \tau^*_0$. Additionally we observe that for higher depths, most of these alternate drift paths lead to a sizeable shift between the received and hypothesized sequences, making them appear random with respect to each other. Thus, we can reasonably assume that all branch metrics in $\tau^*_0$ beyond depth 1 can be described by $\zeta'$. With this assumption and (19), we attempt to bound the number of visits to nodes in $\tau^*_0$.

$$\sum_{\nu^*_0 \in \tau^*_0} \mathbb{E} \left[ \left( \frac{\mu(\nu^*_0) - y}{\Delta} + 1 \right) \phi(\nu^*_0)|T^*_\min = y \right]$$

$$= 2^b \sum_{\nu^*_0 \in \tau^*_0} \sum_{\nu^*_0 \to \nu^*_1 \leq \nu^*_0 \to \nu^*_1} \sum_{z_{1} \geq y} \left( \frac{z_{1} - y}{\Delta} + 1 \right) f_{\nu^*_0}(y, \mu) + \frac{\mu - y}{\Delta} f_{\nu^*_0}(y, \mu)$$

where $b' = b + \log_2 \lambda$. Numerical verification reveals that for $3$If received sequence is error-free, $T^*_\min = 0$. $4$Only few equally likely drift sequences exist, after one insertion or deletion.
reasonable channel parameters, $\mathbb{E}[\epsilon]\leq 0$. This is crucial in ensuring that Fano’s decoder is unlikely to pick the wrong successor at any step. It thus follows that any infinite random walk in $\tau_0$ will eventually fall below a finite $T_{\text{min}}$, or

$$\sum_{t} \sum_{z<y} f_t(y, z) = 1. \quad (21)$$

By reducing (19) and (20) with (21), we may bound (17) as

$$\mathbb{E}(C|y_0)T_{\text{min}} = y \leq C_1 2^{-\sigma_1 y} + C_2 2^{-\sigma_2 y} + C_3 \left( -\frac{y}{\Delta} + 1 \right). \quad (22)$$

where $C_1, C_2$ and $C_3$ are constants for a specific channel and convolutional code parameters, and $\sigma_1 > 0$ satisfies

$$g_1(\sigma_1) = \mathbb{E}[2^{\sigma_1}c'] = 2^{-\sigma_1'}. \quad (23)$$

Upon applying (12) and (22) to (9), we obtain an upper bound on $C_{\text{av}}$ that converges to a finite value when (10) is upheld.

V. Results

To evaluate this decoding strategy, we use three standard, rate $1/3$ convolutional codes with distinct constraint lengths, as outlined in Table I.

| Code | $[c, b, m]$ | Gen. polynomial | $d_{\text{free}}$ |
|------|-------------|----------------|-----------------|
| CC1  | $[3, 1, 1]$ | 1 3 3          | 5               |
| CC2  | $[3, 1, 6]$ | 117 127 155    | 15              |
| CC3  | $[3, 1, 10]$| 3645 2133 3347 | 21              |

The generator polynomials are stated in octal form. We simulated the transmission of terminated codewords with $L = 300$ information blocks and offset by a random sequence$^5$, over a channel with parameters set to $P_t = P_d$ and $P_s = 0$, 0.02 $^6$. To limit decoder complexity, we constrain the maximum allowable insertions/deletions per block to $c$, and ignore drift states of magnitudes exceeding 30. Additionally, if the number

$^5$Helps to maintain synchronization, like marker or watermark codes [12].

$^6$Results for more parameter ranges will be presented in a longer version.

A. Decoding performance

The decoding accuracy of Fano’s sequential decoder is assessed by measuring bit error rates over a range of $P_t = P_d$, and performing a comparison with Viterbi decoder, as depicted in Fig. 3. Since it only partially examines a given code tree, Fano’s decoder is inherently sub-optimal and is thus outperformed by the Viterbi decoder. Unsurprisingly, a higher value of $P_s$ worsens the bit error rate for both decoders.

B. Simulated complexity

The practical complexities of the two decoders are compared in terms of the number of branch metric computations performed. So, for a Viterbi decoder, we compute the total number of branches in the associated trellis, while for Fano’s algorithm, it suffices to measure the average number of forward steps taken, denoted by $\mathcal{F}_{av}$, and to multiply this with the number of outgoing edges per node, as explained in Section IV. For the former case, we limit our attention to the initial non-terminating part of the trellis. Since in either case, a single node produces $2^b(4t_{\text{max}}+d_{\text{max}}+1)$ outgoing branches, we define the following complexity reduction factor,

$$\nu = \frac{N_{\text{tot}}}{\mathcal{F}_{av}} \quad (24)$$

where $N_{\text{tot}}$ denotes the total number of nodes in the trellis.

Fig. 4 exhibits substantial reductions in decoding effort by the use of Fano’s decoder, particularly for low error probabilities. We also note that as channel noise deteriorates, especially beyond $P_t^*$ and $P_s^*$ which mark operation at cutoff rate for $P_t = 0$ and $P_s = 0.02$ respectively, Fano’s decoder gradually loses its computational merit. This is because as noise levels worsen, the decoder is forced to examine many more incorrect paths until the correct one is found.

$^7$A direct comparison with [15] was not possible since we could not extract the data points in its plots with sufficient precision. Furthermore, the trellis employed in [15] differs markedly from the model used here, in that [15] only permits one insertion at each stage and allows edges to skip over a time step. Finally, [12] demonstrated the superiority of the lattice metric used for the decoder metric in (6), over the one defined in [15].
REFERENCES

[1] S. M. H. T. Yazdi, H. M. Kiah, E. Garcia-Ruiz, J. Ma, H. Zhao, and O. Milenkovic, “DNA-based storage: Trends and methods,” IEEE Transactions on Molecular, Biological and Multi-Scale Communications, vol. 1, no. 3, pp. 230–248, Sep. 2015.

[2] D. O. Heckel, G. Mikutis, and R. N. Grass, “A characterization of the DNA data storage channel,” Scientific Reports, vol. 9, no. 9663, Jul. 2019.

[3] V. I. Levenshtein, “Binary codes capable of correcting deletions, insertions and reversals,” Soviet Physics Doklady, vol. 10, no. 8, pp. 707–710, Feb. 1966.

[4] H. Mercier, V. Bhargava, and V. Tarokh, “A survey of error-correcting codes for channels with symbol synchronization errors,” IEEE Communications Surveys & Tutorials, vol. 12, no. 1, pp. 87–96, 2010.

[5] M. C. Davey and D. J. C. MacKay, “Reliable communication over channels with insertions, deletions, and substitutions,” IEEE Transactions on Information Theory, vol. 47, no. 2, pp. 687–698, Feb. 2001.

[6] L. Calabi and W. Hartnett, “Some general results of coding theory with applications to the study of codes for the correction of synchronization errors,” Information and Control, vol. 15, no. 3, pp. 235–249, Sep. 1969.

[7] D. J. Coumou and G. Sharma, “Insertion, Deletion Codes With Feature-Based Embedding: A New Paradigm for Watermark Synchronization With Applications to Speech Watermarking,” IEEE Transactions on Information Forensics and Security, vol. 3, no. 2, pp. 153–165, Jun. 2008.

[8] J. A. Briffa, V. Buttigieg, and S. Wesemeyer, “Time-varying block codes for synchronisation errors: Maximum a posteriori decoder and practical issues,” The Journal of Engineering, vol. 2014, no. 6, pp. 340–351, Jun. 2014.

[9] R. G. Gallager, “Sequential decoding for binary channel with noise and synchronization errors,” Lincoln Lab Group, Arlington, VA, USA, Tech. Rep., Sep. 1961.

[10] M. Rahmati and T. M. Duman, “Bounds on the Capacity of Random Insertion and Deletion-Additive Noise Channels,” IEEE Transactions on Information Theory, vol. 59, no. 9, pp. 5534–5546, Sep. 2013.

[11] M. F. Mansour and A. H. Tewfik, “Convolutional decoding in the presence of synchronization errors,” IEEE Journal on Selected Areas in Communications, vol. 28, no. 2, pp. 218–227, Feb. 2010.

[12] V. Buttigieg and N. Farrugia, “Improved bit error rate performance of convolutional codes with synchronization errors,” in Proc. Int. Conf. Comm., London, Jun. 2015, pp. 4077–4082.

[13] A. Lenz, I. Maarouf, L. Welser, A. Wachter-Zeh, E. Rosnes, and A. Graell i Amat, “Concatenated Codes for Recovery From Multiple Reads of DNA Sequences,” in Proc. Inf. Theory Workshop, Riva del Garda, Italy, Apr. 2021, pp. 1–5.

[14] J. M. Wozencraft, “Sequential decoding for reliable communication,” Massachusetts Institute of Technology, Tech. Rep., Aug. 1957, p. 182.

[15] M. F. Mansour and A. H. Tewfik, “Convolutional codes for channels with substitutions, insertions, and deletions,” in Proc. Global Commun. Conf., vol. 2, Taipei, Taiwan: IEEE, 2002, pp. 1051–1055.

[16] L. Bahl and F. Jelinek, “Decoding for channels with insertions, deletions, and substitutions with applications to speech recognition,” IEEE Transactions on Information Theory, vol. 21, no. 4, pp. 404–411, Jul. 1975.

[17] R. Johannesson and K. S. Zigangirov, “Sequential decoding,” in Fundamentals of Convolutional Coding. Hoboken, New Jersey: John Wiley & Sons, Inc., 2015, pp. 425–484.

[18] R. Fano, “A heuristic discussion of probabilistic decoding,” IEEE Transactions on Information Theory, vol. 9, no. 2, pp. 64–74, Apr. 1963.

[19] J. Massey, “Variable-length codes and the Fano metric,” IEEE Transactions on Information Theory, vol. 18, no. 1, pp. 196–198, Jan. 1972.

[20] R. Pemantle and M. C. Wilson, Analytic Combinatorics in Several Variables, ser. Cambridge Studies in Advanced Mathematics 140. Cambridge: Cambridge University Press, 2013.

[21] S. Melczer and M. C. Wilson, “Asymptotics of lattice walks via analytic combinatorics in several variables,” Discrete Mathematics & Theoretical Computer Science, vol. DMTCS Proceedings, 28th... P. 6390, Apr. 2020.

[22] S. Melczer, An Invitation to Analytic Combinatorics: From One to Several Variables, ser. Texts & Monographs in Symbolic Computation. Cham: Springer International Publishing, 2021.

[23] R. G. Gallager, Information theory and reliable communication. New York, NY: Wiley, 1986.