On Low-Complexity Maximum-Likelihood Decoding of Convolutional Codes

Jie Luo, *Member, IEEE*

***Abstract*—This letter considers the average complexity of maximum-likelihood (ML) decoding of convolutional codes. ML decoding can be modeled as finding the most probable path taken through a Markov graph. Integrated with the Viterbi algorithm (VA), complexity reduction methods often use the sum log likelihood (SLL) of a Markov path as a bound to disprove the optimality of other Markov path sets and to consequently avoid exhaustive path search. In this letter, it is shown that SLL-based optimality tests are inefficient if one fixes the coding memory and takes the codeword length to infinity. Alternatively, optimality of a source symbol at a given time index can be testified using bounds derived from log likelihoods of the neighboring symbols. It is demonstrated that such neighboring log likelihood (NLL)-based optimality tests, whose efficiency does not depend on the codeword length, can bring significant complexity reduction. The results are generalized to ML sequence detection in a class of discrete-time hidden Markov systems.**

***Index Terms*—Coding complexity, convolutional code, hidden Markov model, maximum-likelihood (ML) decoding, Viterbi algorithm (VA).**

I. Introduction

Forney showed that maximum-likelihood (ML) decoding of convolutional codes is equivalent to finding the most probable path taken through a Markov graph [1]. Denote the codeword length by N and the coding memory by ν. For each time index, the number of Markov states in the Markov graph is exponential in ν. The total number of Markov states is, therefore, exponential in ν but linear in N. Define the complexity of a decoder as the number of visited Markov states normalized by the codeword length N. Practical ML decoding is often achieved using the Viterbi algorithm (VA) [1], whose complexity does not scale in N but scales exponentially in ν. Well-known decoders such as the list decoders, the sequential decoders, and the iterative decoders are able to achieve near optimal error performance with low average complexity. However, these decoders do not guarantee the output of the ML codeword [2]. If obtaining the ML codeword is strictly enforced, to avoid exhaustive path search, the decoder must develop certain optimality test criterion (OTC) [3] to test whether the ML path belongs to a Markov path set.

Two major OTCs have been used in the ML decoding of convolutional codes. The first one is the "path covering criterion" (PCC) [4] used in the VA [1]. VA visits *all* Markov states in chronological order [1]. For each time index, the decoder maintains a set of "cover" Markov paths each passing one of the Markov states [1]. According to the PCC, the "cover" Markov path passing a Markov state disproves the optimality of all other Markov paths passing the same state. The second OTC is the SLL-based OTCs used extensively in the sphere decoder [6], [5]. Sphere decoder models ML decoding as finding the lattice point closest to the channel output in the signal space [5]. Hence, the distance between the channel output and an arbitrary lattice point upper bounds the distance from the channel output to the ML codeword. Such distance bound is based on the SLL of the corresponding codeword, and is used in the sphere decoder [6], [5] as well as other ML decoders [3] as the key means to avoid exhaustive codeword search.

Manuscript received December 22, 2006; revised March 01, 2008. Current version published November 21, 2008. The material in this correspondence was presented in part at the IEEE 2007 International Conference on Acoustics, Speech, and Signal Processing, Honolulu, HI, April 2007.

The author is with the Electrical and Computer Engineering Department, Colorado State University, Fort Collins, CO 80523 USA (e-mail: rockey@engr.colostate.edu).

Communicated by E. Viterbo, Associate Editor for Coding Techniques.
Digital Object Identifier 10.1109/TIT.2008.2006385

Assume PCC-based optimality test is always implemented. In this letter, we first show that *additional* complexity reduction brought by the SLL-based optimality test diminishes as one fixes the coding memory ν and takes the codeword length N to infinity. We then show whether the ML message contains a particular symbol at a given time index can be tested using an OTC that depends only on the log likelihood of channel output symbols in a *fixed-sized* time neighborhood. We call such test the NLL-based optimality test, and show its efficiency does not depend on the codeword length. We theoretically demonstrate that NLL-based optimality test can bring significant complexity reduction to ML decoding when the communication system has a high signal-to-noise ratio (SNR). Complexity of the decoder using SLL-base optimality test, on the other hand, remains the same as the VA for all SNRs if the codeword length is taken to infinity. The results are also generalized to ML sequence detection in a class of discrete-time hidden Markov systems.

II. Problem Formulation

Let C be an (n,k) convolutional code over GF(q) defined by a minimal [7] polynomial generator matrix $\boldsymbol{G}(D)$ [7]

$$\boldsymbol{G}(D) = \boldsymbol{G}[0] + \boldsymbol{G}[1]D + \cdots + \boldsymbol{G}[\nu-1]D^{\nu-1}$$

where D is the delay operator; $\boldsymbol{G}[l]$, $0 \le l < \nu$, are $k \times n$ matrices over GF(q).

Denote the message by a *sequence* of vector *symbols*

$$\boldsymbol{x}(D) = \boldsymbol{x}[d]D^d + \boldsymbol{x}[d+1]D^{d+1} + \dots$$

where d is the time index, possibly negative; $\boldsymbol{x}[d]$, $\forall d$, are row vectors of dimension k over GF(q). The corresponding codeword is given by

$$\boldsymbol{y}(D) = \boldsymbol{x}(D)\boldsymbol{G}(D) = \sum_d \sum_{l=0}^{\nu-1} \boldsymbol{x}[d-l]\boldsymbol{G}[l]D^d.$$

We assume $\boldsymbol{x}[d] = \boldsymbol{0}$ for $d < 0$ and $d \ge N$. We term N the codeword length. Define a function $g_q(y)$ that maps y from GF(q) to $\mathcal{R}$ (the set of real numbers) in one-to-one sense. If $\boldsymbol{y}(D)$ is a vector sequence, $g_q(\boldsymbol{y}(D))$ applies the mapping to each element of $\boldsymbol{y}(D)$. Assume the codeword is transmitted over a memoryless Gaussian channel. The channel output is given by

$$\begin{aligned}\boldsymbol{r}(D) &= g_q(\boldsymbol{y}(D)) + \boldsymbol{n}(D)\\ &= g_q(\boldsymbol{x}(D)\boldsymbol{G}(D)) + \boldsymbol{n}(D)\end{aligned}$$

where $\boldsymbol{n}(D) = \boldsymbol{n}[d]D^d + \boldsymbol{n}[d+1]D^{d+1} + \cdots$ is the noise sequence with $\boldsymbol{n}[d] \sim N(\boldsymbol{0}, \sigma^2\boldsymbol{I})$ being i.i.d. Gaussian. Define the scaled SNR of the system as $\mathrm{SNR} = \frac{1}{\sigma^2}$. In Section V, we show that the results are generalizable not only to other channel models, but also to a class of hidden Markov systems.

Given the channel output, for any message $\boldsymbol{x}(D)$ and its corresponding codeword $\boldsymbol{y}(D) = \boldsymbol{x}(D)\boldsymbol{G}(D)$, we define the "negative SLL" as

$$S_x(\boldsymbol{x}(D)) = S_y(\boldsymbol{y}(D)) = \sum_{d=0}^{N+\nu-1} \|\boldsymbol{r}[d] - g_q(\boldsymbol{y}[d])\|^2.$$

The objective is to find the ML message $\boldsymbol{x}_{\mathrm{ML}}(D)$

$$\boldsymbol{x}_{\mathrm{ML}}(D) = \underset{\boldsymbol{x}[d], 0 \le d < N}{\operatorname{argmin}} S_x(\boldsymbol{x}(D)).$$

III. Inefficiency of Sum Log Likelihood-Based Optimality Test

For ML decoders using SLL-based optimality test, the decoder first obtains a quick guess of the message without solving the ML decoding problem [3]. SLL of the obtained message is then used to help disproving the optimality of certain Markov path sets. We make an ideal assumption that the "guessed" message equals the transmitted message, which is denoted by $\boldsymbol{x}(D)$, and $\boldsymbol{y}(D) = \boldsymbol{x}(D)\boldsymbol{G}(D)$. The corresponding negative SLL is given by

$$S_x(\boldsymbol{x}(D)) = \sum_{d=0}^{N+\nu-1} \|\boldsymbol{r}[d] - g_d(\boldsymbol{y}[d])\|^2 = \sum_{d=0}^{N+\nu-1} \|\boldsymbol{n}[d]\|^2.$$

Consider a subset of time indices $D_d^x \subseteq [0, N)$. Let $\{\tilde{\boldsymbol{x}}[d] | d \in D_d^x\}$ be a *partial message* defined only at D_d^x. Denote by $\{\tilde{\boldsymbol{x}}(D_d^x)\}$ the set of messages satisfying

$$\{\tilde{\boldsymbol{x}}(D_d^x)\} = \{\boldsymbol{x}_0(D) | \boldsymbol{x}_0[d] = \tilde{\boldsymbol{x}}[d], \forall d \in D_d^x, \boldsymbol{x}_0(D) \neq \boldsymbol{x}(D)\}.$$

Suppose the decoder wants to test whether it can disprove the optimality of $\{\tilde{\boldsymbol{x}}(D_d^x)\}$, i.e., whether $\boldsymbol{x}_{\mathrm{ML}}(D) \notin \{\tilde{\boldsymbol{x}}(D_d^x)\}$. A common practice [3] is to find a lower bound, $S_x^L(\tilde{\boldsymbol{x}}(D_d^x))$, such that

$$S_x(\boldsymbol{x}_0(D)) \geq S_x^L(\tilde{\boldsymbol{x}}(D_d^x)), \quad \forall\, \boldsymbol{x}_0(D) \in \{\tilde{\boldsymbol{x}}(D_d^x)\}.$$

If the lower bound $S_x^L(\tilde{\boldsymbol{x}}(D_d^x))$ is larger than $S_x(\boldsymbol{x}(D))$, then we have $S_x(\boldsymbol{x}_0(D)) \geq S_x^L(\tilde{\boldsymbol{x}}(D_d^x)) > S_x(\boldsymbol{x}(D))$ for all $\boldsymbol{x}_0(D) \in \{\tilde{\boldsymbol{x}}(D_d^x)\}$, which means $\boldsymbol{x}_{\mathrm{ML}}(D) \notin \{\tilde{\boldsymbol{x}}(D_d^x)\}$.

We skip the proof that the SLL lower bounds appeared in the literature satisfy the following assumption.

Assumption 1: Given $\{\tilde{\boldsymbol{x}}(D_d^x)\}$, let $D_d^y \subseteq [0, N+\nu)$ be the maximum time index set, over which we can find a *partial codeword* $\tilde{\boldsymbol{y}}(D_d^y)$ such that for all $\boldsymbol{x}_0(D) \in \{\tilde{\boldsymbol{x}}(D_d^x)\}$ with $\boldsymbol{y}_0(D) = \boldsymbol{x}_0(D)\boldsymbol{G}(D)$, we have $\boldsymbol{y}_0[d] = \tilde{\boldsymbol{y}}[d]$ for all $d \in D_d^y$. We also have $|D_d^y| \leq |D_d^x| + \nu$. We assume the existence of a positive constant $\epsilon \in (0, 1]$, whose value does *not* depend on N, such that

$$S_x^L(\tilde{\boldsymbol{x}}(D_d^x)) \leq \sum_{d \in D_d^y} \|\boldsymbol{r}[d] - g_q(\tilde{\boldsymbol{y}}[d])\|^2 + (N + \nu - |D_d^y|)(1-\epsilon)n\sigma^2.$$

As demonstrated in [3], if we fix N, using $S_x^L(\tilde{\boldsymbol{x}}(D_d^x)) > S_x(\boldsymbol{x}(D))$ as the OTC to disprove the optimality of message set $\{\tilde{\boldsymbol{x}}(D_d^x)\}$ can bring significant complexity reduction to ML decoding, especially under high SNR. However, if we define $D_e \subseteq D_d^y$ as the subset of time indices corresponding to the erroneous codeword symbols, i.e.

$$D_e = \{d | d \in D_d^y, \tilde{\boldsymbol{y}}(d) \neq \boldsymbol{y}(d)\} \quad (1)$$

the following proposition shows that SLL-based optimality tests become inefficient if $N - |D_d^x|$ is taken to infinity while $|D_e|$ is kept finite.

Lemma 1: Assume the generator matrix $\boldsymbol{G}(D)$ is fixed, and, therefore, the constraint length ν is fixed. Consider message sets characterized by $\{\tilde{\boldsymbol{x}}(D_d^x)\}$ for arbitrary D_d^x under the constraint of a fixed D_e, where $D_e \subseteq D_d^y$ is defined in (1) and the derivation of D_d^y is specified in Assumption 1. If we fix SNR and take $N - |D_d^x| \to \infty$

$$\lim_{N-|D_d^x| \to \infty} P\{S_x^L(\tilde{\boldsymbol{x}}(D_d^x)) > S_x(\boldsymbol{x}(D))\} = 0.$$

If we first take $N - |D_d^x| \to \infty$ and then take $\mathrm{SNR} \to \infty$

$$\lim_{\mathrm{SNR}\to\infty} \lim_{N-|D_d^x| \to \infty} P\{S_x^L(\tilde{\boldsymbol{x}}(D_d^x)) > S_x(\boldsymbol{x}(D))\} = 0. \quad (2)$$

Proof: Since $|D_d^y| \leq |D_d^x| + \nu$, $N - |D_d^x| \to \infty$ implies $N - |D_d^y| \to \infty$. According to Assumption 1, we have

$$\begin{aligned}
\frac{S_x^L(\tilde{\boldsymbol{x}}(D_d^x)) - S_x(\boldsymbol{x}(D))}{N + \nu - |D_d^y|}
&\leq \frac{1}{N+\nu-|D_d^y|}\left(\sum_{d\in D_e} \|\boldsymbol{r}[d] - g_q(\tilde{\boldsymbol{y}}[d])\|^2\right) \\
&\quad + (1-\epsilon)n\sigma^2 - \frac{1}{N+\nu-|D_d^y|}\left(\sum_{d\in D_e} \|\boldsymbol{n}[d]\|^2\right) \\
&\quad - \frac{1}{N+\nu-|D_d^y|}\sum_{d\notin D_d^y} \|\boldsymbol{n}[d]\|^2. \quad (3)
\end{aligned}$$

Denote the right-hand side (RHS) of (3) by U_0, we have with probability one, $\lim_{N-|D_d^y|\to\infty} U_0 = -\epsilon n\sigma^2 < 0$. Hence

$$\begin{aligned}
&\lim_{N-|D_d^x|\to\infty} P\left\{S_x^L(\tilde{\boldsymbol{x}}(D_d^x)) > S_x(\boldsymbol{x}(D))\right\} \\
&= \lim_{N-|D_d^y|\to\infty} P\left\{\frac{S_x^L(\tilde{\boldsymbol{x}}(D_d^x)) - S_x(\boldsymbol{x}(D))}{N+\nu-|D_d^y|} > 0\right\} \\
&\leq \lim_{N-|D_d^y|\to\infty} P\{U_0 > 0\} = 0. \quad (4)
\end{aligned}$$

Since (4) holds for all SNR, it remains true if we first take $N - |D_d^x|$ to infinity and then take SNR to infinity.[1] □

With the help of Lemma 1, inefficiency of SLL-based optimality tests is characterized by the following lemma, the proof of which is skipped.

Lemma 2: Let C_{sll} be the complexity of an ML decoder that only uses PCC- and SLL-based optimality tests for complexity reduction. Let C_{va} be the complexity of the Viterbi decoder, in which, only PCC-based optimality test is used. For any $\delta > 0$, we have

$$\lim_{N\to\infty} P\{C_{\mathrm{sll}} \geq (1-\delta)C_{va}\} = 1$$
$$\lim_{\mathrm{SNR}\to\infty} \lim_{N\to\infty} P\{C_{\mathrm{sll}} \geq (1-\delta)C_{va}\} = 1.$$

IV. Neighboring Log Likelihood-Based Optimality Test

We first propose in Theorem 1 a class of NLL-based optimality tests.

Theorem 1: Define $d_{\min}^2$, $d_{\max}^2$ by

$$d_{\min}^2 = \min_{\boldsymbol{y}_1 \neq \boldsymbol{y}_2} \|g_q(\boldsymbol{y}_1) - g_q(\boldsymbol{y}_2)\|^2$$
$$d_{\max}^2 = \max_{\boldsymbol{y}_1 \neq \boldsymbol{y}_2} \|g_q(\boldsymbol{y}_1) - g_q(\boldsymbol{y}_2)\|^2$$

where $\boldsymbol{y}_1$, $\boldsymbol{y}_2$ are n-dimensional vectors over $GF(q)$. Let ξ be an arbitrary constant, M be an arbitrary integer

$$0 < \xi < \frac{d_{\min}^2}{2}, \quad M > \frac{\nu d_{\max}^2}{3\xi}. \quad (5)$$

[1] Note that the order in which limits are taken in (2) is important. If we fix N and take SNR to infinity first, we can get

$$\lim_{N-|D_d^x|\to\infty} \lim_{\mathrm{SNR}\to\infty} P\{S_x^L(\tilde{\boldsymbol{x}}(D_d^x)) > S_x(\boldsymbol{x}(D))\} = 1.$$

Let $\boldsymbol{x}_0(D)$ be a message whose corresponding codeword is $\boldsymbol{y}_0(D)$. For any time index m, if the following inequality is satisfied for all $d \in [m-2M\nu, m+2M\nu)$

$$\|\boldsymbol{r}[d] - g_q(\boldsymbol{y}_0[d])\| < \frac{d_{\min}^2}{2} - \xi \tag{6}$$

and the following inequalities hold:

$$\sum_{d=m+2M\nu}^{m+(2M+1)\nu-1} \|\boldsymbol{r}[d] - g_q(\boldsymbol{y}_0[d])\|^2 \le M\xi - \nu d_{\max}^2$$
$$\sum_{d=m-(2M+1)\nu}^{m-2M\nu-1} \|\boldsymbol{r}[d] - g_q(\boldsymbol{y}_0[d])\|^2 \le M\xi - \nu d_{\max}^2 \tag{7}$$

then we must have $\boldsymbol{x}_0[\tilde{m}] = \boldsymbol{x}_{\rm ML}[\tilde{m}], \forall \tilde{m} \in [m, m+\nu)$.

Theorem 1 is implied by Theorem 3 in Section V.

Note that, as long as $g_q()$ and ν are given, the values of ξ and M can be fixed, e.g., $\xi = \frac{d_{\min}^2}{4}$ and $M = \lceil \frac{4\nu d_{\max}^2}{3d_{\min}^2} \rceil$. Given M, the optimality test presented in Theorem 1 testifies the optimality of $\{\boldsymbol{x}[\tilde{m}] | \tilde{m} \in [m, m+\nu)\}$ using the log likelihood of channel output symbols within a *fixed-sized* time interval $[m-(2M+1)\nu, m+(2M+1)\nu)$. It is quite intuitive to see, efficiency of the test does not depend on the codeword length if all other parameters are fixed.

Efficiency of the OTC proposed in Theorem 1 is characterized by the following lemma.

Lemma 3: Assume ξ and M are chosen to satisfy (5). Let m be an arbitrary time index. Let $\boldsymbol{y}_0(D)$ equal the transmitted codeword within time interval $[m-(2M+1)\nu, m+(2M+1)\nu)$. Define OPT_m as the event that (7) is satisfied and (6) holds for all $d \in [m-2M\nu, m+2M\nu)$. Fix all other parameters and take SNR $\to \infty$ yields

$$\lim_{\text{SNR}\to\infty} P\{\text{OPT}_m\} = 1. \tag{8}$$

The same conclusion holds if we first take N to infinity

$$\lim_{\text{SNR}\to\infty} \lim_{N\to\infty} P\{\text{OPT}_m\} = 1. \tag{9}$$

Proof: If $\boldsymbol{y}_0(D)$ equals the transmitted codeword within time interval $[m-(2M+1)\nu, m+(2M+1)\nu)$, for $d \in [m-(2M+1)\nu, m+(2M+1)\nu)$, we have $\boldsymbol{r}[d] - g_q(\boldsymbol{y}_0[d]) = \boldsymbol{n}[d]$. Consequently, (8) and (9) hold because $\|\boldsymbol{n}[d]\|^2$ are i.i.d. χ^2, whose mean and variance converge to 0 as SNR$\to\infty$. □

Lemma 3 implies, if there is a suboptimal decoder whose probability of *symbol* detection error (as opposed to sequence detection error) is low under high SNR, then NLL-based optimality tests can help transforming the suboptimal detector to an ML detector with only marginal increase in average decoding complexity. Such transformation can be achieved by the following three-step ML decoding framework.

- Step 1: The decoder uses a suboptimal algorithm (denoted by $\Phi_{\rm sub}$) to obtain a quick guess of the codeword $\tilde{\boldsymbol{y}}(D)$ and its corresponding message $\tilde{\boldsymbol{x}}(D)$.
- Step 2: An NLL-based optimality test (specified in Theorem 1) is applied to each symbol of $\tilde{\boldsymbol{x}}(D)$. The decoder maintains a source symbol set sequence $X(D)$, with $X[d]$ being the source symbol set of time index d. If $\tilde{\boldsymbol{x}}[d] = \boldsymbol{x}_{\rm ML}[d]$ can be confirmed by the optimality test, we let $X[d] = \{\tilde{\boldsymbol{x}}[d]\}$; otherwise, we let $X[d]$ be the set of all possible source symbol vectors at time index d.
- Step 3: The decoder uses a modified VA to search for the ML source message. The modified VA visits a Markov state only if all source symbols corresponding to the Markov state belong to the source symbol sets $X[d]$ of the corresponding time indices.

Implementing the modified VA is quite straightforward. Hence, its further description is skipped.

Theorem 2: Let $P_e\{\Phi_{\rm sub}\}$ be the probability of *symbol* detection error of $\Phi_{\rm sub}$. Assume, while fixing all other parameters,

$$\lim_{\text{SNR}\to\infty} P_e\{\Phi_{\rm sub}\} = 0, \quad \lim_{\text{SNR}\to\infty} \lim_{N\to\infty} P_e\{\Phi_{\rm sub}\} = 0. \tag{10}$$

Let $C_{\rm mva}$ be the average number of Markov states per time unit visited by the modified VA in the third step of the ML decoder. For any $\delta > 0$, we have

$$\lim_{\text{SNR}\to\infty} P\{C_{\rm mva} \le 1+\delta\} = 1$$
$$\lim_{\text{SNR}\to\infty} \lim_{N\to\infty} P\{C_{\rm mva} \le 1+\delta\} = 1.$$

Proof: Let $\boldsymbol{x}(D)$, $\boldsymbol{y}(D)$ be the actual source message and the transmitted codeword, respectively. Let $\tilde{\boldsymbol{x}}(D)$, $\tilde{\boldsymbol{y}}(D)$ be the message and the codeword output by $\Phi_{\rm sub}$. According to (10), for any time index m, we have

$$\lim_{\text{SNR}\to\infty} P\left\{ \begin{array}{c} \tilde{\boldsymbol{y}}[d] = \boldsymbol{y}[d], \\ \forall d \in [m-2(M-1)\nu, m+(2M+1)\nu) \end{array} \right\} = 1. \tag{11}$$

where M is the parameter of the NLL-based optimality test specified in Theorem 1. According to (11), Lemma 2, and Theorem 1, for any m, if $\tilde{\boldsymbol{y}}[d] = \boldsymbol{y}[d], \forall d \in [m-2(M-1)\nu, m+(2M+1)\nu)$, then the probability that the NLL-based optimality test can confirm $\tilde{\boldsymbol{x}}[d] = \boldsymbol{x}_{\rm ML}[d], \forall d \in [m, m+\nu)$ converges to one as SNR $\to\infty$. Consequently, letting $X[d]$ be the source symbol set maintained by the ML decoder in the second step, we have

$$\lim_{\text{SNR}\to\infty} P\{|X[d]| = 1, \forall d \in [m, m+\nu)\} = 1, \quad \forall m. \tag{12}$$

Since the worst case complexity of the modified VA is bounded, (12) implies, for any $\delta > 0$, $\lim_{\text{SNR}\to\infty} P\{C_{\rm mva} \le 1+\delta\} = 1$. Since all derivations hold if we first take $N \to \infty$, we also have $\lim_{\text{SNR}\to\infty} \lim_{N\to\infty} P\{C_{\rm mva} \le 1+\delta\} = 1$. □

Note that the three steps of the ML decoder can be implemented in parallel in the sense that each step can process some of the source symbols without waiting for the previous step to *completely* finish its work [8].

V. Maximum-Likelihood Sequence Detection in a Class of Hidden Markov Systems

In this section, we generalize the results of Section IV to ML sequence detection (MLSD) in a class of first order discrete-time hidden Markov systems.

Let $\boldsymbol{u}(D) = \boldsymbol{u}[d]D^d + \boldsymbol{u}[d+1]D^{d+1} + \cdots$ be a first-order Markov sequence, where $\boldsymbol{u}[d]$ represents the Markov state (at time d), which is a k_ν-dimensional row vector defined over GF(q). We assume $\boldsymbol{u}[d] = \boldsymbol{0}$ for $d < 0$ and $d \ge N$, with N being the sequence length. Define $\boldsymbol{y}[d] = \boldsymbol{y}(\boldsymbol{u}[d])$ as the "processed state," which is a *deterministic* function of $\boldsymbol{u}[d]$. $\boldsymbol{y}[d]$ is a n-dimensional row vector defined over $GF(q)$. We term $\boldsymbol{y}(D) = \boldsymbol{y}[d]D^d + \boldsymbol{y}[d+1]D^{d+1} + \cdots$ the processed state sequence. Let $\boldsymbol{r}(D) = \boldsymbol{r}[d]D^d + \boldsymbol{r}[d+1]D^{d+1} + \cdots$ be the observation sequence, where $\boldsymbol{r}[d]$ is a n-dimensional row vector with real-valued elements.

Denote the state transition probability of the hidden Markov system by

$$P_t(\boldsymbol{u}_1|\boldsymbol{u}_2) = P\{\boldsymbol{u}[d+1] = \boldsymbol{u}_1 | \boldsymbol{u}[d] = \boldsymbol{u}_2\}.$$

Define the transition probability ratio bound p_{tr} by

$$p_{tr} = \min_{\substack{\boldsymbol{u}_1, \boldsymbol{u}_2, P_t(\boldsymbol{u}_1|\boldsymbol{u}_2)>0 \\ \boldsymbol{u}_3, \boldsymbol{u}_4, P_t(\boldsymbol{u}_3|\boldsymbol{u}_4)>0}} \frac{P_t(\boldsymbol{u}_1|\boldsymbol{u}_2)}{P_t(\boldsymbol{u}_3|\boldsymbol{u}_4)}.$$

We assume the Markov chain is ergodic and homogeneous. Therefore, there exists a positive integer ν, such that

$$P\{\boldsymbol{u}[d+\nu] = \boldsymbol{u}_1 | \boldsymbol{u}[d] = \boldsymbol{u}_2\} \ne 0, \quad \forall \boldsymbol{u}_1, \boldsymbol{u}_2. \tag{13}$$

Denote the observation distribution function by

$$F_o(\boldsymbol{r}|\boldsymbol{y}_1) = P\{\boldsymbol{r}[d] \leq \boldsymbol{r}|\boldsymbol{y}[d] = \boldsymbol{y}_1\}.$$

Let the corresponding probability density function (or probability mass function) be $f_o(\boldsymbol{r}|\boldsymbol{y}_1)$.

We also make the following two key assumptions.

Assumption 2: We assume state processing $\boldsymbol{y}[d] = \boldsymbol{y}(\boldsymbol{u}[d])$ does not compromise the observability of the Markov states in the sense that there exists a positive integer ν satisfying the following property. Given two Markov state sequences $\boldsymbol{u}(D)$ and $\tilde{\boldsymbol{u}}(D)$. For any time index d, if $\boldsymbol{u}[d] \neq \tilde{\boldsymbol{u}}[d]$, then we can find a time index $m \in (d-\nu, d+\nu)$, such that $\boldsymbol{y}(\boldsymbol{u}[m]) \neq \boldsymbol{y}(\tilde{\boldsymbol{u}}[m])$.

Note that it is valid to assume the same constant ν in (13) and in Assumption 2.

Assumption 3: Assume the existence of two functions: $L_l(\boldsymbol{r}, \boldsymbol{y}_1)$ and $L_u(\boldsymbol{r}, \boldsymbol{y}_1)$. Assume $L_l(\boldsymbol{r}, \boldsymbol{y}_1)$ and $L_u(\boldsymbol{r}, \boldsymbol{y}_1)$ possess the following two properties. First, the following inequalities hold for all $\boldsymbol{r}$ and $\boldsymbol{y}_1$:

$$\begin{aligned} L_l(\boldsymbol{r}, \boldsymbol{y}_1) &\leq \min_{\boldsymbol{y}_2, \boldsymbol{y}_2 \neq \boldsymbol{y}_1} [-\log(f_o(\boldsymbol{r}|\boldsymbol{y}_2)) + \log(f_o(\boldsymbol{r}|\boldsymbol{y}_1))] \\ L_u(\boldsymbol{r}, \boldsymbol{y}_1) &\geq \max_{\boldsymbol{y}_2 \neq \boldsymbol{y}_3} [-\log(f_o(\boldsymbol{r}|\boldsymbol{y}_2)) + \log(f_o(\boldsymbol{r}|\boldsymbol{y}_3))]. \end{aligned} \tag{14}$$

Second, the complexity of evaluating $L_l(\boldsymbol{r}, \boldsymbol{y}_1)$ and $L_u(\boldsymbol{r}, \boldsymbol{y}_1)$ is low in the sense that they do not require the search of any processed state other than $\boldsymbol{y}_1$.

Given the observation sequence $\boldsymbol{r}(D)$, the negative SLL of a state sequence $\boldsymbol{u}(D)$ is obtained by

$$S_u(\boldsymbol{u}(D)) = -\sum_{d=0}^{N} \log(f_o(\boldsymbol{r}[d]|\boldsymbol{y}[d]) P_t(\boldsymbol{u}[d]|\boldsymbol{u}[d-1])).$$

The objective of MLSD is to find the ML sequence that minimizes the negative SLL

$$\boldsymbol{u}_{\mathrm{ML}}(D) = \underset{\boldsymbol{u}[d], 0 \leq d < N}{\operatorname{argmin}} S_u(\boldsymbol{u}(D)).$$

The following theorem gives a class of NLL-based optimality tests.

Theorem 3: Assume the discrete-time Markov system satisfies Assumptions 2 and 3. Let $\rho > 0$ be a positive constant. Given a Markov state sequence $\boldsymbol{u}(D)$ and the corresponding processed states $\boldsymbol{y}(D)$. Let p_{tr} be defined by (13). For any time index m, if there is an integer $M > 0$ such that for all $d \in [m - 2M\nu, m + 2M\nu)$

$$L_l(\boldsymbol{r}[d], \boldsymbol{y}[d]) > 3\nu(\rho - \log p_{\mathrm{tr}}), \tag{15}$$

and

$$\sum_{d=m-(2M+1)\nu}^{m-2M\nu-1} L_u(\boldsymbol{r}, \boldsymbol{y}[d]) \leq 3M\nu\rho + \nu \log p_{\mathrm{tr}}$$

then $\boldsymbol{u}[m+\nu-1] = \boldsymbol{u}_{\mathrm{ML}}[m+\nu-1]$ must be true.

The Proof of Theorem 3 is given in Appendix A.

For communication systems following a discrete-time hidden Markov model, $f_o(\boldsymbol{r}|\boldsymbol{y}_1)$ often belongs to an ensemble of density (or probability) functions, with the actual realization determined by the SNR. In words, $f_o(\boldsymbol{r}|\boldsymbol{y}_1, \mathrm{SNR})$ is a function of the SNR. Assume the discrete-time Markov system satisfies Assumption 3, where both functions $L_l(\boldsymbol{r}, \boldsymbol{y}_1)$ and $L_u(\boldsymbol{r}, \boldsymbol{y}_1)$ can be functions of the SNR. We make the following assumption.

Assumption 4: Assume the observation density (or probability) $f_o(\boldsymbol{r}|\boldsymbol{y}_1, \mathrm{SNR})$ is a function of the SNR. Assume the discrete-time Markov system satisfies Assumption 3. Let the actual state sequence and the processed state sequence be $\boldsymbol{u}(D)$ and $\boldsymbol{y}(D)$, respectively. Define two positive numbers $d^2_{\min}$ and $d^2_{\max}$ as follows:

$$\begin{aligned} \frac{d^2_{\min}}{2} &= \sup\left\{\gamma \geq 0; \lim_{\mathrm{SNR}\to\infty} P\{L_l(\boldsymbol{r}[d], \boldsymbol{y}[d]) \geq \gamma \mathrm{SNR}\} = 1\right\} \\ d^2_{\max} &= \inf\left\{\gamma \geq 0; \lim_{\mathrm{SNR}\to\infty} P\{L_u(\boldsymbol{r}[d], \boldsymbol{y}[d]) \leq \gamma \mathrm{SNR}\} = 1\right\}. \end{aligned}$$

We assume $d^2_{\min} > 0$, $d^2_{\max} < \infty$.

The following lemma characterizes the efficiency of the OTC proposed in Theorem 3.

Lemma 4: Assume the discrete-time Markov system satisfies Assumptions 2 and 4. Let the state sequence be $\boldsymbol{u}(D)$. Let ξ be an arbitrary constant, M be an arbitrary integer, satisfying

$$0 < \xi < \frac{d^2_{\min}}{2}, \quad M > \frac{\nu d^2_{\max}}{\xi}. \tag{16}$$

Let $\rho = \frac{\xi \mathrm{SNR}}{3\nu}$. Given an arbitrary time index m, define OPT_m as the event that (16) is satisfied and (15) is satisfied for all $d \in [m - 2M\nu, m + 2M\nu)$. If we fix all other parameters except the SNR, we have

$$\lim_{\mathrm{SNR}\to\infty} P\{\mathrm{OPT}_m\} = 1. \tag{17}$$

If we fix all other parameters except the SNR and the sequence length N, we have

$$\lim_{\mathrm{SNR}\to\infty} \lim_{N\to\infty} P\{\mathrm{OPT}_m\} = 1. \tag{18}$$

The Proof of Lemma 4 is skipped.

Note that in Lemma 4, when we take N and SNR to infinity, M can be fixed at a constant. This indicates that, when testing the optimality of a Markov state at a given time index, the NLL-based optimality test only uses observation symbols in a fixed-sized time neighborhood. Based on Theorem 3 and Lemma 3, a three-step ML sequence detector similar to the one presented in Section IV can be developed to transform a suboptimal sequence detector to a low complexity ML sequence detector.

APPENDIX

A. Proof of Theorem 3

Proof: Let $\tilde{\boldsymbol{u}}(D)$ be an arbitrary Markov state sequence with corresponding processed state sequence being $\tilde{\boldsymbol{y}}(D)$. Assume

$$\tilde{\boldsymbol{u}}[m+\nu-1] \neq \boldsymbol{u}[m+\nu-1]. \tag{19}$$

Theorem 3 holds if we can prove that any $\tilde{\boldsymbol{u}}(D)$ satisfying (19) cannot be the ML state sequence.

Let k denote a positive integer. Define two integers K_l and K_r as follows:

$$\begin{aligned} K_l &= \underset{k>0}{\operatorname{argmin}}\{\tilde{\boldsymbol{u}}[m+\nu-1-k\nu] \\ &= \boldsymbol{u}[m+\nu-1-k\nu]\} \\ K_r &= \underset{k>0}{\operatorname{argmin}}\{\tilde{\boldsymbol{u}}[m+\nu-1+k\nu] \\ &= \boldsymbol{u}[m+\nu-1+k\nu]\}. \end{aligned}$$

We consider the following four cases based on the values of K_l and K_r.

Case 1: $K_l \leq 2M+1$, $K_r \leq 2M-1$.
Case 2: $K_l \leq 2M+1$, $K_r > 2M-1$.
Case 3: $K_l > 2M+1$, $K_r \leq 2M-1$.

Case 4: $K_l > 2M+1$, $K_r > 2M-1$.

It can be shown that, for all cases, $\tilde{\boldsymbol{u}}(D)$ cannot be the ML sequence. Since the proofs of the four cases are not essentially different, we only present the proof for Case 4.

We construct a Markov state sequence $\boldsymbol{u}_c(D)$ as follows:

$$\boldsymbol{u}_c[d] = \boldsymbol{u}[d], \quad \text{for } m - 2M\nu \le d < m + 2M\nu$$
$$\boldsymbol{u}_c[d] = \tilde{\boldsymbol{u}}[d], \quad \text{for } d \ge m + (2M+1)\nu$$
$$\boldsymbol{u}_c[d] = \tilde{\boldsymbol{u}}[d], \quad \text{for } d < m - (2M+1)\nu.$$

Let the processed state sequence corresponding to $\boldsymbol{u}_c(D)$ be $\boldsymbol{y}_c(D)$.

Since $\tilde{\boldsymbol{u}}[m+\nu-1+k\nu] \neq \boldsymbol{u}_c[m+\nu-1+k\nu]$ for all $-2M-1 \le k \le 2M-1$, according to Assumption 2, $\tilde{\boldsymbol{y}}(D)$ and $\boldsymbol{y}(D)$ differ at no less than $\lfloor \frac{4M+1}{2} \rfloor$ time indices in the time interval $[m-2M\nu, m+2M\nu)$. According to (14) and (15), we have

$$-\sum_{d=m-2M\nu}^{m+2M\nu-1} \log \frac{f_o(\boldsymbol{r}[d]|\tilde{\boldsymbol{y}}[d])P_t(\tilde{\boldsymbol{u}}[d]|\tilde{\boldsymbol{u}}[d-1])}{f_o(\boldsymbol{r}[d]|\boldsymbol{y}_c[d])P_t(\boldsymbol{u}_c[d]|\boldsymbol{u}_c[d-1])} > \left\lfloor \frac{4M+1}{2} \right\rfloor 3\nu(\rho - \log p_{tr}) + 4M\nu \log p_{tr} \ge 6M\nu\rho.$$

Meanwhile, it is easily seen that the following inequalities hold:

$$-\sum_{d=m+2M\nu}^{m+(2M+1)\nu} \log \frac{f_o(\boldsymbol{r}[d]|\tilde{\boldsymbol{y}}[d])P_t(\tilde{\boldsymbol{u}}[d]|\tilde{\boldsymbol{u}}[d-1])}{f_o(\boldsymbol{r}[d]|\boldsymbol{y}_c[d])P_t(\boldsymbol{u}_c[d]|\boldsymbol{u}_c[d-1])} \ge -\sum_{d=m+2M\nu}^{m+(2M+1)\nu-1} L_u(\boldsymbol{r}[d], \boldsymbol{y}[d]) + (\nu+1)\log p_{tr} \ge -3M\nu\rho,$$

$$-\sum_{d=m-(2M+1)\nu}^{m-2M\nu-1} \log \frac{f_o(\boldsymbol{r}[d]|\tilde{\boldsymbol{y}}[d])P_t(\tilde{\boldsymbol{u}}[d]|\tilde{\boldsymbol{u}}[d-1])}{f_o(\boldsymbol{r}[d]|\boldsymbol{y}_c[d])P_t(\boldsymbol{u}_c[d]|\boldsymbol{u}_c[d-1])} \ge -\sum_{d=m-(2M+1)\nu}^{m-2M\nu-1} L_u(\boldsymbol{r}[d], \boldsymbol{y}[d]) + \nu \log p_{tr} \ge -3M\nu\rho.$$

Consequently

$$-\sum_{d=m-(2M+1)\nu}^{m+(2M+1)\nu} \log \frac{f_o(\boldsymbol{r}[d]|\tilde{\boldsymbol{y}}[d]))P_t(\tilde{\boldsymbol{u}}[d]|\tilde{\boldsymbol{u}}[d-1])}{f_o(\boldsymbol{r}[d]|\boldsymbol{y}_c[d])P_t(\boldsymbol{u}_c[d]|\boldsymbol{u}_c[d-1])} > -3M\nu\rho - 3M\nu\rho + 6M\nu\rho = 0. \quad (20)$$

(20) implies that $\boldsymbol{u}_c(D)$ "covers" $\tilde{\boldsymbol{u}}(D)$. According to the PCC [4], $\tilde{\boldsymbol{u}}(D)$ cannot be the ML sequence.

REFERENCES

[1] G. Forney, "The Viterbi algorithm," *Proc. IEEE*, vol. 61, no. 3, pp. 268–278, Mar. 1973.

[2] R. Johannesson and K. Zigangirov, *Fundamentals of Convolutional Coding*. Piscataway, NJ: IEEE, 1999.

[3] P. Swaszek and W. Jones, "How often is hard-decision decoding enough?," *IEEE Trans. Inf. Theory*, vol. 44, pp. 1187–1193, May 1998.

[4] M. Ariel and J. Snyders, "Error-trellises for convolutional codes-part II: Decoding methods," *IEEE Trans. Commun.*, vol. 47, pp. 1015–1024, Jul. 1999.

[5] B. Hassibi and H. Vikalo, "On the sphere decoding algorithm I: Expected complexity," *IEEE Trans. Signal Process.*, vol. 53, no. 8, pp. 2806–2818, Aug. 2005.

[6] U. Fincke and M. Pohst, "Improved methods for calculating vectors of short length in a lattice, including a complexity analysis," *Math. Comput.*, vol. 44, pp. 463–471, Apr. 1985.

[7] G. Forney, "Structural analysis of convolutional codes via dual codes," *IEEE Trans. Inf. Theory*, vol. IT-19, pp. 512–518, Jul. 1973.

[8] J. Luo, "Fast maximum likelihood sequence detection over vector intersymbol interference channels," in *Proc. IEEE ICASSP*, Honolulu, HI, Apr. 2007.

Clifford Code Constructions of Operator Quantum Error-Correcting Codes

Andreas Klappenecker, *Member, IEEE*, and Pradeep Kiran Sarvepalli

***Abstract*—Recently, operator quantum error-correcting codes have been proposed to unify and generalize decoherence free subspaces, noiseless subsystems, and quantum error-correcting codes. This correspondence introduces a natural construction of such codes in terms of Clifford codes, an elegant generalization of stabilizer codes due to Knill. Character-theoretic methods are used to derive a simple method to construct operator quantum error-correcting codes from any classical additive code over a finite field, which obviates the need for self-orthogonal codes.**

***Index Terms*—Clifford codes, operator quantum error-correcting codes, quantum codes, stabilizer codes, subsystem codes.**

I. INTRODUCTION

One of the main challenges in quantum information processing is the protection of the quantum information against various sources of errors. A possible remedy is given by encoding the quantum information in a subspace C of the state space H of the quantum system. If such a quantum error-correcting code C is well chosen, then many errors can be corrected through active recovery operations. A more recent development is the encoding of quantum information into a subsystem A of the state space [13], [14]. This means that C is further decomposed into a tensor product of vector spaces A and B such that

$$H = C \oplus C^{\perp} = (A \otimes B) \oplus C^{\perp}.$$

One refers to C as an operator quantum error-correcting code with subsystem A and co-subsystem B. Some authors refer to the co-subsystem as the gauge subsystem. One advantage is that errors affecting the co-subsystem B alone do not require any active error-correction. Furthermore, one can detect all errors that map the encoded information into the orthogonal complement $C^{\perp}$ of C.

The operator quantum error-correcting codes generalize and unify the main methods of passive and active quantum error-correction: decoherence free subspaces, noiseless subsystems, and quantum error-correcting codes. More background on operator quantum error-correcting codes can be found, for example, in [2], [11], [13], [14], [12], and [15].

Manuscript received December 22, 2006; revised March 01, 2008. Current version published November 21, 2008. This work was supported in part by the National Science Foundation (NSF) under Grant CCF-0622201, by the NSF CAREER Award CCF-0347310, and by a TITF project.

The authors are with the Department of Computer Science, Texas A&M University, College Station, TX 77843 USA (e-mail: klappi@cs.tamu.edu; pradeep@cs.tamu.edu).

Communicated by A. Winter, Associate Editor for Quantum Information Theory.

Digital Object Identifier 10.1109/TIT.2008.2006385