

# On Low Complexity Maximum Likelihood Decoding of Convolutional Codes

Jie Luo, *Member, IEEE*

## Abstract

This paper 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 such as the sphere decoder 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 paper, 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 to ML decoding of convolutional codes. 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 decoding, Viterbi algorithm

## I. INTRODUCTION

We study the algorithms that reduce the average complexity of maximum likelihood (ML) decoding of convolutional codes. By ML decoding, we mean the decoder uses code-search to find, and to guarantee the output of, the most likely codeword.

Forney showed that 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  $\nu$ . For each time index, the number of Markov states in the Markov graph is exponential in  $\nu$ . The total number of Markov states is therefore exponential in  $\nu$  but linear in  $N$ . Define the complexity of

The author is with the Electrical and Computer Engineering Department, Colorado State University, Fort Collins, CO 80523.  
E-mail: rocky@engr.colostate.edu.

This work was supported by National Science Foundation grant CCF-0728826.

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) [2][1], whose complexity does not scale in  $N$  but scales exponentially in  $\nu$ . Well known decoders such as the list decoders [3], the sequential decoders [4], and the iterative decoders [5] are able to achieve near optimal error performance with low average complexity. However, these decoders do not guarantee the output of the ML codeword [6].

If obtaining the ML codeword is strictly enforced (see Section VII for justification), to avoid exhaustive path search, the decoder must develop certain criterion or bound that can be used to disprove the optimality of a Markov path set. This is equivalent to developing an optimality test criterion (OTC) [7] to test whether the ML path (or codeword) belongs to the complementary path set (or codeword set)<sup>1</sup>.

Two major OTCs have been used in the ML decoding of convolutional codes. The first one is the “path covering criterion” (PCC) (explained in [8] and in Appendix A) used in the VA [2][1]. VA visits *all* Markov states in chronological order [1]. For each time index, the decoder maintains a set of “cover” (defined in Appendix A) 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 sum log likelihood (SLL)-based OTCs used extensively in the sphere decoder [10][9]. Sphere decoder models ML decoding as finding the lattice point closest to the channel output in the signal space [9]. 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 [10][9] as well as other ML decoders [7] as the key means to avoid exhaustive codeword search. In [11][12], Vikalo and Hassibi showed that PCC-based and SLL-based optimality tests can be combined to find the ML codeword without visiting all Markov states.

Assume PCC-based optimality test is always implemented. In this paper, we first show that *additional* complexity reduction brought by the SLL-based optimality test diminishes as one fixes the coding memory  $\nu$  and takes the codeword length  $N$  to infinity. Such inefficiency is due to the fact that SLL-based OTC does not exploit the structure of the convolutional code. Searching the ML codeword is equivalent to finding the ML source message, which contains a sequence of source symbols. We 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-*

<sup>1</sup>In the literature such as [7], OTC refers to a criterion designed to test whether a single codeword is optimum. In this paper, we extend the definition of OTC to a general criterion that can either verify or disprove the optimality of a codeword set.

sized time neighborhood. We call such test the neighboring log likelihood (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 SNR 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 [13].

## II. PROBLEM FORMULATION

Let  $C$  be an  $(n, k)$  convolutional code over  $\text{GF}(q)$  defined by a polynomial generator matrix  $\mathbf{G}(D)$  [14],

$$\mathbf{G}(D) = \mathbf{G}[0] + \mathbf{G}[1]D + \dots + \mathbf{G}[\nu - 1]D^{\nu-1}, \quad (1)$$

where  $D$  is the delay operator;  $\nu$  is the coding memory;  $\mathbf{G}[l]$ ,  $l = 0, \dots, \nu - 1$ , are  $k \times n$  matrices over  $\text{GF}(q)$ . Assume  $\mathbf{G}(D)$  is a minimal encoder [14].

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

$$\mathbf{x}(D) = \mathbf{x}[d]D^d + \mathbf{x}[d + 1]D^{d+1} + \dots, \quad (2)$$

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

$$\mathbf{y}(D) = \mathbf{x}(D)\mathbf{G}(D) = \sum_d \sum_{l=0}^{\nu-1} \mathbf{x}[d - l]\mathbf{G}[l]D^d. \quad (3)$$

To simplify the presentation, we assume time index  $d$  takes all integer values. We assume  $\mathbf{x}[d] = \mathbf{0}$  for  $d < 0$  and  $d \geq N$ . We term  $N$  the codeword length.

Define a function  $g_q(y)$  that maps  $y$  from  $\text{GF}(q)$  to  $\mathcal{R}$  (the set of real numbers) in one-to-one sense. If  $\mathbf{y}(D)$  is a vector sequence,  $g_q(\mathbf{y}(D))$  applies the mapping to each of the elements of  $\mathbf{y}(D)$ , respectively<sup>2</sup>. Assume the codeword is transmitted over a memoryless Gaussian channel. The channel output symbol sequence is given by

$$\mathbf{r}(D) = g_q(\mathbf{y}(D)) + \mathbf{n}(D) = g_q(\mathbf{x}(D)\mathbf{G}(D)) + \mathbf{n}(D), \quad (4)$$

where  $\mathbf{n}(D) = \mathbf{n}[d]D^d + \mathbf{n}[d + 1]D^{d+1} + \dots$  is the noise sequence with  $\mathbf{n}[d] \sim N(\mathbf{0}, \sigma^2 \mathbf{I})$  being i.i.d. Gaussian. Without loss of generality, we define the scaled signal to noise ratio of the system as  $\text{SNR} = \frac{1}{\sigma^2}$ . In Section VI, we show that the results are generalizable not only to other channel models, but also to a class of hidden Markov systems.

<sup>2</sup>Hence the output of  $g_q(\mathbf{y}(D))$  is a vector sequence of the same length and dimension as  $\mathbf{y}(D)$ .

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

$$S_x(\mathbf{x}(D)) = S_y(\mathbf{y}(D)) = \sum_{d=0}^{N+\nu-1} \|\mathbf{r}[d] - g_q(\mathbf{y}[d])\|^2. \quad (5)$$

The objective of ML decoding is to find the ML message  $\mathbf{x}_{ML}(D)$  that minimizes the negative SLL,

$$\mathbf{x}_{ML}(D) = \underset{\mathbf{x}[d], 0 \leq d < N}{\operatorname{argmin}} S_x(\mathbf{x}(D)). \quad (6)$$

Throughout this paper, we assume PCC-based optimality test is always implemented. For the sake of completeness, a description of PCC-based optimality test is given in Appendix A.

### 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 source message without solving the ML decoding problem. SLL of the obtained message is then used to help disproving the optimality of certain Markov path sets and consequently to avoid exhaustive path search. We make an ideal assumption that the “guessed” message equals the transmitted message<sup>3</sup>. We show in this section that, even under this ideal assumption, complexity reduction brought by the SLL-based optimality tests still diminishes as we take  $N$  to infinity.

Let  $\mathbf{x}(D)$  be the actual source message, which is also the message “guessed” by the decoder. Let  $\mathbf{y}(D) = \mathbf{x}(D)\mathbf{G}(D)$  be the transmitted codeword. The corresponding negative SLL is given by

$$S_x(\mathbf{x}(D)) = \sum_{d=0}^{N+\nu-1} \|\mathbf{r}[d] - g_d(\mathbf{y}[d])\|^2 = \sum_{d=0}^{N+\nu-1} \|\mathbf{n}[d]\|^2. \quad (7)$$

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

$$\{\tilde{\mathbf{x}}(D_d^x)\} = \{\mathbf{x}_0(D) | \mathbf{x}_0[d] = \tilde{\mathbf{x}}[d], \forall d \in D_d^x, \mathbf{x}_0(D) \neq \mathbf{x}(D)\}. \quad (8)$$

Suppose the decoder wants to test whether it can disprove the optimality of  $\{\tilde{\mathbf{x}}(D_d^x)\}$ , i.e., whether  $\mathbf{x}_{ML}(D) \notin \{\tilde{\mathbf{x}}(D_d^x)\}$ . A common practice [7][11][12] is to find a lower bound, denoted by  $S_x^L(\tilde{\mathbf{x}}(D_d^x))$ , of the negative SLLs of the messages in  $\{\tilde{\mathbf{x}}(D_d^x)\}$ .

$$S_x(\mathbf{x}_0(D)) \geq S_x^L(\tilde{\mathbf{x}}(D_d^x)), \quad \forall \mathbf{x}_0(D) \in \{\tilde{\mathbf{x}}(D_d^x)\}. \quad (9)$$

If the lower bound  $S_x^L(\tilde{\mathbf{x}}(D_d^x))$  is larger than  $S_x(\mathbf{x}(D))$  obtained in (7), then we have  $S_x(\mathbf{x}_0(D)) \geq S_x^L(\tilde{\mathbf{x}}(D_d^x)) > S_x(\mathbf{x}(D))$  for all  $\mathbf{x}_0(D) \in \{\tilde{\mathbf{x}}(D_d^x)\}$ , which means the ML message is not in  $\{\tilde{\mathbf{x}}(D_d^x)\}$ .

<sup>3</sup>Note that the decoder still needs to testify whether the guessed message is indeed the ML solution. If it is not, then a search for the ML message must be carried out.

In Appendix B, we show that the SLL lower bounds appeared in the literature satisfy the following assumption.

**Assumption 1:** Given  $\{\tilde{\mathbf{x}}(D_d^x)\}$ , let  $D_d^y \subseteq [0, N + \nu]$  be the maximum time index set, over which we can find a *partial codeword*  $\tilde{\mathbf{y}}(D_d^y)$  such that for all  $\mathbf{x}_0(D) \in \{\tilde{\mathbf{x}}(D_d^x)\}$  with  $\mathbf{y}_0(D) = \mathbf{x}_0(D)\mathbf{G}(D)$ , we have  $\mathbf{y}_0[d] = \tilde{\mathbf{y}}[d]$  for all  $d \in D_d^y$ . Note that  $D_d^y$  and  $\tilde{\mathbf{y}}(D_d^y)$  are uniquely determined by  $\{\tilde{\mathbf{x}}(D_d^x)\}$ . 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

$$S_x^L(\tilde{\mathbf{x}}(D_d^x)) \leq \sum_{d \in D_d^y} \|\mathbf{r}[d] - g_q(\tilde{\mathbf{y}}[d])\|^2 + (N + \nu - |D_d^y|)(1 - \epsilon)n\sigma^2. \quad (10)$$

As demonstrated in [11][7], if we fix  $N$ , using  $S_x^L(\tilde{\mathbf{x}}(D_d^x)) > S_x(\mathbf{x}(D))$  as the OTC to disprove the optimality of message set  $\{\tilde{\mathbf{x}}(D_d^x)\}$  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.,

$$D_e = \{d | d \in D_d^y, \tilde{\mathbf{y}}(d) \neq \mathbf{y}(d)\}, \quad (11)$$

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  $\mathbf{G}(D)$  is fixed, and therefore the constraint length  $\nu$  is fixed. Consider message sets characterized by  $\{\tilde{\mathbf{x}}(D_d^x)\}$  for arbitrary  $D_d^x$  but under the constraint of a fixed  $D_e$ , where  $D_e \subseteq D_d^y$  is defined in (11) and the derivation of  $D_d^y$  is specified in Assumption 1.

If we fix SNR and take  $N - |D_d^x|$  to infinity, we have

$$\lim_{N - |D_d^x| \rightarrow \infty} P\{S_x^L(\tilde{\mathbf{x}}(D_d^x)) > S_x(\mathbf{x}(D))\} = 0. \quad (12)$$

If we first take  $N - |D_d^x|$  to infinity and then take SNR to infinity, we have

$$\lim_{\text{SNR} \rightarrow \infty} \lim_{N - |D_d^x| \rightarrow \infty} P\{S_x^L(\tilde{\mathbf{x}}(D_d^x)) > S_x(\mathbf{x}(D))\} = 0. \quad (13)$$

*Proof:* Since  $|D_d^y| \leq |D_d^x| + \nu$ , taking  $N - |D_d^x|$  to infinity implies taking  $N - |D_d^y|$  to infinity.

According to Assumption 1, we have

$$\begin{aligned} \frac{S_x^L(\tilde{\mathbf{x}}(D_d^x)) - S_x(\mathbf{x}(D))}{N + \nu - |D_d^y|} &\leq \frac{1}{N + \nu - |D_d^y|} \left( \sum_{d \in D_e} \|\mathbf{r}[d] - g_q(\tilde{\mathbf{y}}[d])\|^2 \right) + (1 - \epsilon)n\sigma^2 \\ &\quad - \frac{1}{N + \nu - |D_d^y|} \left( \sum_{d \in D_e} \|\mathbf{n}[d]\|^2 \right) - \frac{1}{N + \nu - |D_d^y|} \sum_{d \notin D_e} \|\mathbf{n}[d]\|^2. \end{aligned} \quad (14)$$

Since  $\mathbf{n}[d]$  are i.i.d. Gaussian with covariance matrix  $\sigma^2 \mathbf{I}$ ,  $\|\mathbf{n}[d]\|^2$  are i.i.d.  $\chi^2$  with mean  $n\sigma^2$  and variance  $2n\sigma^4$ . Therefore  $\frac{1}{N+\nu-|D_d^y|} \sum_{d \notin D_d^y} \|\mathbf{n}[d]\|^2 \rightarrow n\sigma^2$ ,  $\frac{1}{N+\nu-|D_d^y|} \left( \sum_{d \in D_e} \|\mathbf{r}[d] - g_q(\tilde{\mathbf{y}}[d])\|^2 \right) \rightarrow 0$ , and  $\frac{1}{N+\nu-|D_d^y|} \sum_{d \in D_e} \|\mathbf{n}[d]\|^2 \rightarrow 0$  with probability one as  $N - |D_d^y| \rightarrow \infty$ . Consequently, denote the right hand side of (14) by  $U_0$ , we have with probability one,

$$\lim_{N-|D_d^y| \rightarrow \infty} U_0 = -\epsilon n \sigma^2 < 0. \quad (15)$$

This yields

$$\begin{aligned} \lim_{N-|D_d^x| \rightarrow \infty} P \left\{ S_x^L(\tilde{\mathbf{x}}(D_d^x)) > S_x(\mathbf{x}(D)) \right\} &= \lim_{N-|D_d^y| \rightarrow \infty} P \left\{ \frac{S_x^L(\tilde{\mathbf{x}}(D_d^x)) - S_x(\mathbf{x}(D))}{N + \nu - |D_d^y|} > 0 \right\} \\ &\leq \lim_{N-|D_d^y| \rightarrow \infty} P \{ U_0 > 0 \} = 0. \end{aligned} \quad (16)$$

Since (16) holds for all SNR, the conclusion remains true if we take SNR to infinity after  $N - |D_d^x|$  is taken to infinity<sup>4</sup>. ■

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

**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,

$$\begin{aligned} \lim_{N \rightarrow \infty} P \{ C_{sll} \geq (1 - \delta) C_{va} \} &= 1 \\ \lim_{\text{SNR} \rightarrow \infty} \lim_{N \rightarrow \infty} P \{ C_{sll} \geq (1 - \delta) C_{va} \} &= 1. \end{aligned} \quad (17)$$

■

The proof of Lemma 2 is given in Appendix C.

#### IV. NEIGHBORING LOG LIKELIHOOD-BASED OPTIMALITY TEST

We propose in Theorem 1 a class of NLL-based optimality tests, whose efficiency does not depend on the codeword length  $N$ . We show in Section V that these NLL-based optimality tests can significantly reduce the average complexity of ML decoding under high SNR. This is in contrast to the inefficiency of SLL-based optimality tests which are not able to bring meaningful complexity reduction if  $N$  is taken to infinity first.

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

$$d_{\min}^2 = \min_{\mathbf{y}_1 \neq \mathbf{y}_2} \|g_q(\mathbf{y}_1) - g_q(\mathbf{y}_2)\|^2, \quad d_{\max}^2 = \max_{\mathbf{y}_1 \neq \mathbf{y}_2} \|g_q(\mathbf{y}_1) - g_q(\mathbf{y}_2)\|^2, \quad (18)$$

<sup>4</sup>Note that the order in which limits are taken in (13) is important. If we fix  $N$  and take SNR to infinity first, we can get  $\lim_{N-|D_d^x| \rightarrow \infty} \lim_{\text{SNR} \rightarrow \infty} P \{ S_x^L(\tilde{\mathbf{x}}(D_d^x)) > S_x(\mathbf{x}(D)) \} = 1$ .

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

$$0 < \xi < \frac{d_{\min}^2}{2}, \quad M > \frac{\nu d_{\max}^2}{3\xi}. \quad (19)$$

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

$$\|\mathbf{r}[d] - g_q(\mathbf{y}_0[d])\| < \frac{d_{\min}^2}{2} - \xi, \quad (20)$$

and the following inequalities hold,

$$\begin{aligned} \sum_{d=m+2M\nu}^{m+(2M+1)\nu-1} \|\mathbf{r}[d] - g_q(\mathbf{y}_0[d])\|^2 &\leq M\xi - \nu d_{\max}^2 \\ \sum_{d=m-(2M+1)\nu}^{m-2M\nu-1} \|\mathbf{r}[d] - g_q(\mathbf{y}_0[d])\|^2 &\leq M\xi - \nu d_{\max}^2, \end{aligned} \quad (21)$$

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

We skip the proof of Theorem 1 since the result is implied by Theorem 3 presented in Section VI.

Note that the values of  $d_{\min}$  and  $d_{\max}$  only depend on the  $g_q(\cdot)$  function. Hence, as long as  $g_q(\cdot)$  and  $\nu$  are given, the values of  $\xi$  and  $M$  can be fixed, e.g.,  $\xi = \frac{d_{\min}^2}{4}$  and  $M = \left\lceil \frac{4\nu d_{\max}^2}{3d_{\min}^2} \right\rceil$ . Given  $M$ , the optimality test presented in Theorem 1 testifies the optimality of  $\{\mathbf{x}[\tilde{m}] | \tilde{m} \in [m, m + \nu]\}$  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  $\xi$  and  $M$  are chosen to satisfy (19). Let  $m$  be an arbitrary time index. Let  $\mathbf{y}_0(D)$  equal the transmitted codeword within time interval  $[m - (2M + 1)\nu, m + (2M + 1)\nu]$ . Define  $\text{OPT}_m$  as the event that (21) is satisfied and (20) is satisfied for all  $d \in [m - 2M\nu, m + 2M\nu]$ .

Fix all other parameters and take SNR to infinity, we have

$$\lim_{\text{SNR} \rightarrow \infty} P\{\text{OPT}_m\} = 1. \quad (22)$$

The same conclusion holds if we first take  $N$  to infinity, then take SNR to infinity.

$$\lim_{\text{SNR} \rightarrow \infty} \lim_{N \rightarrow \infty} P\{\text{OPT}_m\} = 1. \quad (23)$$

*Proof:* If  $\mathbf{y}_0(D)$  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

$$\mathbf{r}[d] - g_q(\mathbf{y}_0[d]) = \mathbf{n}[d]. \quad (24)$$

Consequently, (22) and (23) hold because  $\|\mathbf{n}[d]\|^2$  are i.i.d.  $\chi^2$ , whose mean,  $\frac{n}{\text{SNR}}$ , and variance,  $\frac{2n}{\text{SNR}^2}$ , converge to 0 as SNR goes to infinity. ■

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. An example of such transformation is presented in the following section.

## V. A THREE-STEP ML DECODING FRAMEWORK

The communication system given in Section II follows a discrete-time hidden Markov model [13], where each Markov state at time index  $d$  corresponds to a possible combination of source symbols in time interval  $(d - \nu, d]$ . If a decoder obtains the ML codeword using the VA, all Markov states within time interval  $[\nu, N]$  have to be visited. Alternatively, if one can use a low complexity algorithm to disprove the optimality of most of the Markov states, then the VA can limit its search by visiting only a small subset of Markov states.

Following this idea, the three-step ML decoding framework is given as follows.

- Step 1: The decoder uses a suboptimal algorithm (denoted by  $\Phi_{\text{sub}}$ ) to obtain a quick guess of the codeword  $\tilde{\mathbf{y}}(D)$  and its corresponding source message  $\tilde{\mathbf{x}}(D)$ .
- Step 2: An NLL-based optimality test (specified in Theorem 1) is applied to each of the source symbols of  $\tilde{\mathbf{x}}(D)$ . The decoder maintains a source symbol set sequence  $X(D)$ , with  $X[d]$  being the source symbol set of time index  $d$ . If  $\tilde{\mathbf{x}}[d] = \mathbf{x}_{ML}[d]$  can be confirmed by the optimality test, we let  $X[d] = \{\tilde{\mathbf{x}}[d]\}$ ; 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 only difference between the modified VA and the conventional VA is that, 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. Comparing to the three-step decoding algorithm studied in [7], the key advantage of using an NLL-based optimality test is that the test can be applied to an individual source symbol rather than the whole source message.

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

$$\lim_{\text{SNR} \rightarrow \infty} P_e\{\Phi_{\text{sub}}\} = 0, \quad \lim_{\text{SNR} \rightarrow \infty} \lim_{N \rightarrow \infty} P_e\{\Phi_{\text{sub}}\} = 0. \quad (25)$$

Let  $C_{\text{mva}}$  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

$$\lim_{\text{SNR} \rightarrow \infty} P\{C_{\text{mva}} \leq 1 + \delta\} = 1, \quad \lim_{\text{SNR} \rightarrow \infty} \lim_{N \rightarrow \infty} P\{C_{\text{mva}} \leq 1 + \delta\} = 1. \quad (26)$$

*Proof:* Let  $\mathbf{x}(D)$ ,  $\mathbf{y}(D)$  be the actual source message and the transmitted codeword, respectively. Let  $\tilde{\mathbf{x}}(D)$ ,  $\tilde{\mathbf{y}}(D)$  be the source message and the codeword output by  $\Phi_{\text{sub}}$ . According to (25), for any time index  $m$ , we have

$$\lim_{\text{SNR} \rightarrow \infty} P \left\{ \begin{array}{l} \tilde{\mathbf{y}}[d] = \mathbf{y}[d], \\ \forall d \in [m - 2(M - 1)\nu, m + (2M + 1)\nu] \end{array} \right\} = 1. \quad (27)$$

where  $M$  is the parameter of the NLL-based optimality test, specified in Theorem 1. According to (27), Lemma 2, and Theorem 1, for any  $m$ , if  $\tilde{\mathbf{y}}[d] = \mathbf{y}[d], \forall d \in [m - 2(M - 1)\nu, m + (2M + 1)\nu]$ , then the probability that the NLL-based optimality test can confirm  $\tilde{\mathbf{x}}[d] = \mathbf{x}_{\text{ML}}[d], \forall d \in [m, m + \nu]$  converges to one as  $\text{SNR} \rightarrow \infty$ . Consequently, letting  $X[d]$  be the source symbol set maintained by the ML decoder in the second step, we have

$$\lim_{\text{SNR} \rightarrow \infty} P\{|X[d]| = 1, \forall d \in [m, m + \nu]\} = 1, \quad \forall m \quad (28)$$

Since the worst case complexity of the modified VA is bounded, (28) implies, for any  $\delta > 0$ ,  $\lim_{\text{SNR} \rightarrow \infty} P\{C_{\text{mva}} \leq 1 + \delta\} = 1$ .

Since all derivations hold if we first take  $N$  to infinity, we also have  $\lim_{\text{SNR} \rightarrow \infty} \lim_{N \rightarrow \infty} P\{C_{\text{mva}} \leq 1 + \delta\} = 1$ . ■

By sharing computations among optimality tests, it is easy to see that the complexity of the second step of the ML decoder is equivalent, in order, to visiting one Markov state per time unit. Therefore, if  $\Phi_{\text{sub}}$  satisfies (25), as  $\text{SNR} \rightarrow \infty$ , the complexity of the three-step ML decoder converges to the complexity of  $\Phi_{\text{sub}}$ , which can be significantly lower than the complexity of the VA. Moreover, the three steps of the ML decoder can be implemented in a parallelized manner in the sense that each step can process some of the source symbols without waiting for the previous step to *completely* finish its work. An example of such parallelized implementation can be found in [15, The Simple MLSD Algorithm].

## VI. 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 [13]. We demonstrate in Appendix D that the communication system presented in Section II satisfies the model and the key assumptions given in this section.

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

Denote the state transition probability of the hidden Markov system by

$$P_t(\mathbf{u}_1|\mathbf{u}_2) = P\{\mathbf{u}[d+1] = \mathbf{u}_1 | \mathbf{u}[d] = \mathbf{u}_2\}. \quad (29)$$

Define the transition probability ratio bound  $p_{tr}$  by

$$p_{tr} = \min_{\substack{\mathbf{u}_1, \mathbf{u}_2, P_t(\mathbf{u}_1|\mathbf{u}_2) > 0 \\ \mathbf{u}_3, \mathbf{u}_4, P_t(\mathbf{u}_3|\mathbf{u}_4) > 0}} \frac{P_t(\mathbf{u}_1|\mathbf{u}_2)}{P_t(\mathbf{u}_3|\mathbf{u}_4)}. \quad (30)$$

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

$$P\{\mathbf{u}[d+\nu] = \mathbf{u}_1 | \mathbf{u}[d] = \mathbf{u}_2\} \neq 0, \quad \forall \mathbf{u}_1, \mathbf{u}_2. \quad (31)$$

Denote the observation distribution function by

$$F_o(\mathbf{r}|\mathbf{y}_1) = P\{\mathbf{r}[d] \leq \mathbf{r} | \mathbf{y}[d] = \mathbf{y}_1\}. \quad (32)$$

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

We also make the following two key assumptions.

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

Note that we used the same constant  $\nu$  in (31) and in Assumption 2. This is valid because if (31) is satisfied for  $\nu = \nu_0$ , then it is also satisfied for all  $\nu \geq \nu_0$ ; similar property applies to Assumption 2. Consequently, if Assumption 2 holds, a common integer  $\nu$  satisfying both (31) and Assumption 2 can always be found.

**Assumption 3:** Assume the existence of two functions:  $L_l(\mathbf{r}, \mathbf{y}_1)$  and  $L_u(\mathbf{r}, \mathbf{y}_1)$ , both are functions of the channel output symbol  $\mathbf{r}$  and the processed state  $\mathbf{y}_1$ . Assume  $L_l(\mathbf{r}, \mathbf{y}_1)$  and  $L_u(\mathbf{r}, \mathbf{y}_1)$  have the following two properties.

First, the following inequalities hold for all  $\mathbf{r}$  and  $\mathbf{y}_1$ .

$$\begin{aligned} L_l(\mathbf{r}, \mathbf{y}_1) &\leq \min_{\mathbf{y}_2, \mathbf{y}_2 \neq \mathbf{y}_1} [-\log(f_o(\mathbf{r}|\mathbf{y}_2)) + \log(f_o(\mathbf{r}|\mathbf{y}_1))] \\ L_u(\mathbf{r}, \mathbf{y}_1) &\geq \max_{\mathbf{y}_2 \neq \mathbf{y}_3} [-\log(f_o(\mathbf{r}|\mathbf{y}_2)) + \log(f_o(\mathbf{r}|\mathbf{y}_3))]. \end{aligned} \quad (33)$$

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

Note that validity of the results presented in this section does not depend on the second property imposed in Assumption 3. However, we still include the property in the assumption since the key motivation of posing Assumption 3 is to use the two functions  $L_l(\mathbf{r}, \mathbf{y}_1)$  and  $L_u(\mathbf{r}, \mathbf{y}_1)$  as tools to avoid exhaustive Markov state search and hence to reduce the complexity of ML decoding. Also note that the right hand side of the second inequality in (33) is not a function of  $\mathbf{y}_1$ . However, the upper bound on the left hand side is a function of a processed state  $\mathbf{y}_1$  since one often needs a “reference state” in order to upper bound the right hand side of (33). Further explanation is given in Appendix D.

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

$$S_u(\mathbf{u}(D)) = - \sum_{d=0}^N \log(f_o(\mathbf{r}[d]|\mathbf{y}[d])P_t(\mathbf{u}[d]|\mathbf{u}[d-1])). \quad (34)$$

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

$$\mathbf{u}_{ML}(D) = \underset{\mathbf{u}[d], 0 \leq d < N}{\operatorname{argmin}} S_u(\mathbf{u}(D)). \quad (35)$$

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  $\mathbf{u}(D)$  and the corresponding processed states  $\mathbf{y}(D)$ . Let  $p_{tr}$  be defined by (30). For any time index  $m$ , if there is an integer  $M > 0$  such that for all  $d \in [m - 2M\nu, m + 2M\nu)$

$$L_l(\mathbf{r}[d], \mathbf{y}[d]) > 3\nu(\rho - \log p_{tr}), \quad (36)$$

and

$$\begin{aligned} \sum_{d=m+2M\nu}^{m+(2M+1)\nu-1} L_u(\mathbf{r}, \mathbf{y}[d]) &\leq 3M\nu\rho + (\nu + 1) \log p_{tr} \\ \sum_{d=m-(2M+1)\nu}^{m-2M\nu-1} L_u(\mathbf{r}, \mathbf{y}[d]) &\leq 3M\nu\rho + \nu \log p_{tr}, \end{aligned} \quad (37)$$

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

The proof of Theorem 3 is given in Appendix E. Note that Theorem 3 implies Theorem 1 if we set the parameters in Theorem 1 at the corresponding values given in Appendix D.

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

**Assumption 4:** Assume the observation density (or probability)  $f_o(\mathbf{r}|\mathbf{y}_1, \text{SNR})$  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  $\mathbf{u}(D)$  and  $\mathbf{y}(D)$ , respectively. Define two positive numbers  $d_{\min}^2$  and  $d_{\max}^2$  as follows

$$\begin{aligned} \frac{d_{\min}^2}{2} &= \sup \left\{ \gamma \geq 0; \lim_{\text{SNR} \rightarrow \infty} P\{L_l(\mathbf{r}[d], \mathbf{y}[d]) \geq \gamma \text{SNR}\} = 1 \right\}, \\ d_{\max}^2 &= \inf \left\{ \gamma \geq 0; \lim_{\text{SNR} \rightarrow \infty} P\{L_u(\mathbf{r}[d], \mathbf{y}[d]) \leq \gamma \text{SNR}\} = 1 \right\}. \end{aligned} \quad (38)$$

We assume

$$d_{\min}^2 > 0, \quad d_{\max}^2 < \infty. \quad (39)$$

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  $\mathbf{u}(D)$ . Let  $\xi$  be an arbitrary constant,  $M$  be an arbitrary integer, satisfying

$$0 < \xi < \frac{d_{\min}^2}{2}, \quad M > \frac{\nu d_{\max}^2}{\xi}. \quad (40)$$

Let  $\rho = \frac{\xi \text{SNR}}{3\nu}$ . Given an arbitrary time index  $m$ , define  $\text{OPT}_m$  as the event that (37) is satisfied and (36) is satisfied for all  $d \in [m - 2M\nu, m + 2M\nu]$ . If we fix all other parameters except the SNR, we have

$$\lim_{\text{SNR} \rightarrow \infty} P\{\text{OPT}_m\} = 1. \quad (41)$$

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

$$\lim_{\text{SNR} \rightarrow \infty} \lim_{N \rightarrow \infty} P\{\text{OPT}_m\} = 1. \quad (42)$$

We skip the proof of Lemma 4 since it is quite straightforward.

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 V can be developed to transform a suboptimal sequence detector to a low complexity ML sequence detector. The detailed discussion is skipped since it does not essentially differ from the one presented in Section V.

## VII. FURTHER DISCUSSIONS

In a practical system, suboptimal decoders such as the belief-propagation-based iterative decoders [5][6] can achieve near optimal error performance with low complexity. It is natural to ask: if suboptimal decoding only causes a negligible performance loss, why one should even bother with enforcing the ML solution? Note that this question does not suggest a default answer since the argument can also be presented in the opposite direction, i.e., if ML decoding only causes a negligible complexity increase, why one should not use an ML decoder? Nevertheless, the purpose of our work is not to participate in the debate whether ML decoding is practically useful. Rather, one should interpret Theorem 2 as, for convolutional codes, the existence of a well-performed low complexity suboptimal algorithm implies that ML decoding can be carried out with a similar complexity under high SNR. More importantly, such conclusion holds irrespective of the codeword length.

Although the efficiency of SLL-based optimality tests does not depend on the codeword length, NLL-based optimality tests are inefficient only when the codeword length is large. Lemma 1 and Theorem 2 suggest that complexity reduction brought by NLL-based optimality tests can be superior to SLL-based optimality tests even for moderate SNR if the codeword length is large enough.

### APPENDIX

#### A. The Path Covering Criterion

Assume the discrete-time hidden Markov model given in Section VI<sup>5</sup>. Given the observation sequence  $\mathbf{r}(D)$ . Let  $\tilde{\mathbf{u}}(D)$  and  $\mathbf{u}(D)$  be two Markov state sequences whose corresponding processed state sequences are  $\tilde{\mathbf{y}}(D)$  and  $\mathbf{y}(D)$ , respectively. If we can find two time indices  $d_1 < d_2$ , such that  $\tilde{\mathbf{u}}[d_1] = \mathbf{u}[d_1]$ ,  $\tilde{\mathbf{u}}[d_2] = \mathbf{u}[d_2]$ , and

$$\sum_{d=d_1+1}^{d_2} \log \frac{f_o(\mathbf{r}[d]|\tilde{\mathbf{y}}[d-1])P_t(\tilde{\mathbf{u}}[d]|\tilde{\mathbf{u}}[d-1])}{f_o(\mathbf{r}[d]|\mathbf{y}[d-1])P_t(\mathbf{u}[d]|\mathbf{u}[d-1])} < 0, \quad (43)$$

we say  $\mathbf{u}(D)$  “covers”  $\tilde{\mathbf{u}}(D)$ .

**Path Covering Criterion:** Markov state sequence  $\tilde{\mathbf{u}}(D)$  cannot be the ML sequence if we can find another state sequence  $\mathbf{u}(D)$  that covers  $\tilde{\mathbf{u}}(D)$ .

The proof of the PCC is skipped since it is quite well known [8].

We say  $\mathbf{u}(D)$  is a “cover” path with respect to Markov states  $\mathbf{u}[d_1]$  and  $\mathbf{u}[d_2]$  at time indices  $d_1 < d_2$  if, among all Markov paths passing  $\mathbf{u}[d_1]$  and  $\mathbf{u}[d_2]$ ,  $\mathbf{u}(D)$  maximizes  $\sum_{d=d_1+1}^{d_2} \log(f_o(\mathbf{r}[d]|\mathbf{y}[d-1])P_t(\mathbf{u}[d]|\mathbf{u}[d-1]))$ . Assume all Markov paths start from  $\mathbf{u}[-1] = \mathbf{0}$ . We say  $\mathbf{u}(D)$  is a “cover” path with respect to Markov state  $\mathbf{u}[d_1]$  at time index  $d_1 > 0$  if, among all Markov paths passing  $\mathbf{u}[d_1]$ ,  $\mathbf{u}(D)$  maximizes  $\sum_{d=1}^{d_1} \log(f_o(\mathbf{r}[d]|\mathbf{y}[d-1])P_t(\mathbf{u}[d]|\mathbf{u}[d-1]))$ .

<sup>5</sup>It is shown in Appendix D that the model is satisfied by the communication system given in Section II.

### B. Examples of SLL-based Optimality Tests Satisfying Assumption 1

In [12][11], when the decoder branches a Markov path at time index  $m < N$ , the branch is characterized by a *partial message*  $\{\tilde{\mathbf{x}}[0], \tilde{\mathbf{x}}[1], \dots, \tilde{\mathbf{x}}[m]\}$ . For any codeword  $\tilde{\mathbf{y}}(D)$  associated to the branch, we have

$$\tilde{\mathbf{y}}[d] = \sum_{l=0}^{\nu-1} \tilde{\mathbf{x}}[d-l] \mathbf{G}[l]. \quad (44)$$

In other words,  $D_d = [0, m]$ . The negative SLL lower bound is given by

$$S_y(\tilde{\mathbf{y}}(D)) = \sum_{d=0}^{N+\nu-1} \|\mathbf{r}[d] - g_q(\tilde{\mathbf{y}}[d])\|^2 \geq \sum_{d=0}^m \left\| \mathbf{r}[d] - g_q \left( \sum_{l=0}^{\nu-1} \tilde{\mathbf{x}}[d-l] \mathbf{G}[l] \right) \right\|^2, \quad (45)$$

which satisfies Assumption 1 with  $\epsilon = 1$ .

In [7], several SLL-based OTCs were presented for decoding block codes. The decoder obtains a first guess  $\mathbf{y}(D)$  of the codeword. A negative SLL lower bound  $S_y^L \leq S_y(\tilde{\mathbf{y}}(D) \neq \mathbf{y}(D))$  is then developed for the codeword set  $\{\tilde{\mathbf{y}}(D) \neq \mathbf{y}(D)\}$ , which corresponds to the case of  $D_d$  being an empty set in the context of Section III.  $\mathbf{y}(D)$  is optimal if the optimality test  $S_y^L > S_y(\mathbf{y}(D))$  gives a positive answer [7].

The lower bounds  $S_y^L$  presented in [7, Section III] satisfy the following inequality,

$$S_y^L \leq \min_{\tilde{\mathbf{y}}(D) \neq \mathbf{y}(D)} \sum_{d=0}^{N+\nu-1} \|g_q(\tilde{\mathbf{y}}[d]) - g_q(\mathbf{y}[d])\|^2 \quad (46)$$

Since the coding constraint is  $\nu$ , we can always find a codeword  $\tilde{\mathbf{y}}(D) \neq \mathbf{y}(D)$  with  $\tilde{\mathbf{y}}(D)$  differing from  $\mathbf{y}(D)$  at no more than  $\nu$  codeword symbols. This implies that the right hand side of (46) can be upper bounded by a constant, denoted by  $U_1$ , which is not a function of  $N$ .

$$S_y^L \leq \min_{\tilde{\mathbf{y}}(D) \neq \mathbf{y}(D)} \sum_{d=0}^{N+\nu-1} \|g_q(\tilde{\mathbf{y}}[d]) - g_q(\mathbf{y}[d])\|^2 \leq U_1 \quad (47)$$

Consequently, given  $\text{SNR} > 0$  and  $0 < \epsilon < 1$ , there exists a constant  $N_0$  such that Assumption 1 is satisfied for  $N > N_0$ .

### C. Proof of Lemma 2

*Proof:* Assume, in searching the ML codeword, the decoder successfully avoided visiting a Markov state specified by  $\{\mathbf{x}_0[d-\nu+1], \dots, \mathbf{x}_0[d]\}$ . This implies that we can find two time index sets,  $D_0^x \subset [d-\nu+1, d]$  and  $D_d^x$ ,  $D_d^x \cap [d-\nu+1, d] = \phi$ , such that the optimality of *all* message sets  $\{\tilde{\mathbf{x}}(D_0^x \cup D_d^x)\}$  with  $\tilde{\mathbf{x}}[\tilde{d}] = \mathbf{x}_0[\tilde{d}]$ ,  $\forall \tilde{d} \in D_0^x$  is disproved. We choose  $D_d^x$  with *the maximum cardinality* while make sure that, in disproving the optimality of  $\{\mathbf{x}_0[d-\nu+1], \dots, \mathbf{x}_0[d]\}$ , the detector visited *all* the Markov states  $\{\tilde{\mathbf{x}}[\tilde{d}-\nu+1], \dots, \tilde{\mathbf{x}}[\tilde{d}]\}$  satisfying  $[\tilde{d}-\nu+1, \tilde{d}] \subseteq D_d^x$ .

According to the definitions of  $D_0^x$  and  $D_d^x$ , the decoder needs to disprove the optimality of a special message set  $\{\mathbf{x}_0(D_0^x \cup D_d^x)\}$  defined by  $\mathbf{x}_0[\tilde{d}] = \mathbf{x}_0[\tilde{d}]$ ,  $\forall \tilde{d} \in D_0^x$  and  $\mathbf{x}_0[\tilde{d}] = \mathbf{x}[\tilde{d}]$ ,  $\forall \tilde{d} \in D_d^x$ .

The definition of  $D_d^x$  also implies that the decoder needs to obtain a lower bound  $S_x^L(\tilde{\mathbf{x}}(D_0^x \cup D_d^x))$  of the negative SLLs of the messages in  $\{\tilde{\mathbf{x}}(D_0^x \cup D_d^x)\}$ . The lower bound  $S_x^L(\tilde{\mathbf{x}}(D_0^x \cup D_d^x))$  should only be a function of the partial message  $\tilde{\mathbf{x}}(D_0^x \cup D_d^x)$ , but should not depend on any source message symbol whose time index is outside  $D_0^x \cup D_d^x$ . However, since the corresponding  $D_e$  (defined in (11)) of  $\{\mathbf{x}_0(D_0^x \cup D_d^x)\}$  satisfies  $|D_e| \leq 2\nu$ , according to Lemma 1, the probability of disproving the optimality of  $\{\mathbf{x}_0(D_0^x \cup D_d^x)\}$  (using SLL-based optimality test) is low if  $N - |D_0^x \cup D_d^x| \gg 2\nu$ .

To make the argument explicit, the fact that the decoder visits *all* Markov states  $\{\tilde{\mathbf{x}}[\tilde{d} - \nu + 1], \dots, \tilde{\mathbf{x}}[\tilde{d}]\}$  with  $[\tilde{d} - \nu + 1, \tilde{d}] \subseteq D_d^x$  implies

$$C_{sll} \geq \frac{|D_d^x| - \nu}{N + \nu} C_{va}. \quad (48)$$

According to Lemma 1, for any positive constant  $\delta > 0$ , if we fix all other parameters and take  $N$  to infinity, we have<sup>6</sup>

$$\lim_{N \rightarrow \infty} P \left\{ \frac{N - |D_d^x| - |D_0^x|}{2\nu} < \frac{\delta}{2\nu} N \right\} = 1. \quad (49)$$

Combining (48) and (49), we get

$$\lim_{N \rightarrow \infty} P \{C_{sll} \geq (1 - \delta)C_{va}\} = 1. \quad (50)$$

Since (50) holds for any fixed SNR, it still holds if we take SNR to infinity after taking  $N$  to infinity, i.e.,

$$\lim_{\text{SNR} \rightarrow \infty} \lim_{N \rightarrow \infty} P \{C_{sll} \geq (1 - \delta)C_{va}\} = 1. \quad (51)$$

■

#### D. The Hidden Markov Model and Its Key Assumptions

In this section, we show the communication system presented in Section II satisfies the discrete-time hidden Markov model and the key assumptions given in Section VI.

Consider a communication system modeled in Section II. Define  $\mathbf{u}[d] = [\mathbf{x}[d - \nu + 1], \dots, \mathbf{x}[d]]$ . It is easy to see  $\mathbf{u}(D)$  is a Markov sequence. The processed state  $\mathbf{y}[d] = \mathbf{y}(\mathbf{u}[d])$  is only a function of the corresponding Markov state. If two Markov states in successive time indices take the form

$$\begin{aligned} \mathbf{u}[d] &= [\tilde{\mathbf{x}}[d - \nu + 1], \dots, \tilde{\mathbf{x}}[d]] \\ \mathbf{u}[d + 1] &= [\tilde{\mathbf{x}}[d - \nu + 2], \dots, \tilde{\mathbf{x}}[d + 1]], \end{aligned} \quad (52)$$

for some  $\tilde{\mathbf{x}}(D)$ , then we have

$$P_t(\mathbf{u}[d + 1] | \mathbf{u}[d]) = \frac{1}{q^k}. \quad (53)$$

<sup>6</sup>An equivalent statement of (49) is, if  $\frac{N - |D_d^x| - |D_0^x|}{2\nu} < \frac{\delta}{2\nu} N$ , as  $N \rightarrow \infty$ , the probability of disproving the optimality of all message sets  $\{\tilde{\mathbf{x}}(D_0^x \cup D_d^x)\}$  with  $\tilde{\mathbf{x}}[\tilde{d}] = \mathbf{x}_0[\tilde{d}]$ ,  $\forall \tilde{d} \in D_0^x$ , using SLL-based optimality test goes to zero.

Otherwise  $P_t(\mathbf{u}[d+1]|\mathbf{u}[d]) = 0$ . According to (30), we have  $p_{tr} = 1$ .

Since  $\mathbf{u}[d] = [\mathbf{x}[d-\nu+1], \dots, \mathbf{x}[d]]$  does not depend on source symbols at time indices  $m \leq d-\nu$ , we know

$$P_t(\mathbf{u}[d]|\mathbf{u}[d-\nu]) \neq 0, \quad \forall \mathbf{u}[d], \mathbf{u}[d-\nu]. \quad (54)$$

The observation density is given by

$$f_o(\mathbf{r}|\mathbf{y}) = \left(\frac{\text{SNR}}{2\pi}\right)^{\frac{n}{2}} \exp\left(-\frac{\text{SNR}}{2}\|\mathbf{r} - g_q(\mathbf{y})\|^2\right). \quad (55)$$

Next, we show Assumption 2 is satisfied. Let  $\mathbf{u}(D)$  and  $\tilde{\mathbf{u}}(D)$  be two Markov state sequences. Let  $\mathbf{x}(D)$  and  $\mathbf{y}(D)$  be the source message and the codeword corresponding to  $\mathbf{u}(D)$ . Let  $\tilde{\mathbf{x}}(D)$  and  $\tilde{\mathbf{y}}(D)$  be the source message and the codeword corresponding to  $\tilde{\mathbf{u}}(D)$ . For a time index  $d$ , if  $\mathbf{u}[d] \neq \tilde{\mathbf{u}}[d]$ , we can find a time index  $m \in (d-\nu, d]$  such that  $\mathbf{x}[m] \neq \tilde{\mathbf{x}}[m]$ . Consequently, according to [14, Corollary 2], we can find a time index  $\tilde{m} \in [m, m+\nu)$ , such that  $\mathbf{y}[\tilde{m}] \neq \tilde{\mathbf{y}}[\tilde{m}]$ . Therefore, Assumption 2 holds because  $\tilde{m} \in (d-\nu, d+\nu)$ .

Let  $d_{\min}^2$  and  $d_{\max}^2$  be defined in Theorem 1. Let  $\mathbf{y}_1 \neq \mathbf{y}_2$  be two arbitrary codeword symbols. We have the following triangle inequalities,

$$\begin{aligned} \|\mathbf{r} - g_q(\mathbf{y}_2)\| &\geq \|g_q(\mathbf{y}_2) - g_q(\mathbf{y}_1)\| - \|\mathbf{r} - g_q(\mathbf{y}_1)\| \\ \|\mathbf{r} - g_q(\mathbf{y}_2)\| &\leq \|g_q(\mathbf{y}_2) - g_q(\mathbf{y}_1)\| + \|\mathbf{r} - g_q(\mathbf{y}_1)\|. \end{aligned} \quad (56)$$

The first inequality in (56) implies

$$\begin{aligned} \min_{\mathbf{y}_2, \mathbf{y}_2 \neq \mathbf{y}_1} [-\log(f_o(\mathbf{r}|\mathbf{y}_2))] + \log(f_o(\mathbf{r}|\mathbf{y}_1)) &= \min_{\mathbf{y}_2, \mathbf{y}_2 \neq \mathbf{y}_1} \left[ \frac{\text{SNR}}{2} (\|\mathbf{r} - g_q(\mathbf{y}_2)\|^2 - \|\mathbf{r} - g_q(\mathbf{y}_1)\|^2) \right] \\ &\geq \frac{\text{SNR}}{2} d_{\min} (d_{\min} - 2\|\mathbf{r} - g_q(\mathbf{y}_1)\|). \end{aligned} \quad (57)$$

The second inequality in (56) implies

$$\begin{aligned} \max_{\mathbf{y}_2 \neq \mathbf{y}_3} [-\log(f_o(\mathbf{r}|\mathbf{y}_2)) + \log(f_o(\mathbf{r}|\mathbf{y}_3))] &= \max_{\mathbf{y}_2 \neq \mathbf{y}_3} \left[ \frac{\text{SNR}}{2} (\|\mathbf{r} - g_q(\mathbf{y}_2)\|^2 - \|\mathbf{r} - g_q(\mathbf{y}_3)\|^2) \right] \\ &\leq \max_{\mathbf{y}_2} \left[ \frac{\text{SNR}}{2} \|\mathbf{r} - g_q(\mathbf{y}_2)\|^2 \right] \\ &\leq \max_{\mathbf{y}_2} \left[ \text{SNR} (\|\mathbf{r} - g_q(\mathbf{y}_1)\|^2 + \|g_q(\mathbf{y}_2) - g_q(\mathbf{y}_1)\|^2) \right] \\ &\leq \text{SNR} (\|\mathbf{r} - g_q(\mathbf{y}_1)\|^2 + d_{\max}^2). \end{aligned} \quad (58)$$

Therefore, Assumption 3 is satisfied by defining

$$\begin{aligned} L_l(\mathbf{r}, \mathbf{y}_1) &= \frac{\text{SNR}}{2} d_{\min} (d_{\min} - 2\|\mathbf{r} - g_q(\mathbf{y}_1)\|) \\ L_u(\mathbf{r}, \mathbf{y}_1) &= \text{SNR} (\|\mathbf{r} - g_q(\mathbf{y}_1)\|^2 + d_{\max}^2). \end{aligned} \quad (59)$$

Note that evaluating  $L_l(\mathbf{r}, \mathbf{y}_1)$  and  $L_u(\mathbf{r}, \mathbf{y}_1)$  does not involve visiting any processed state other than  $\mathbf{y}_1$ .

If  $\mathbf{y}[d]$  and  $\mathbf{r}[d]$  are the actual codeword symbol and the channel output at time index  $d$ ,  $\|\mathbf{r}[d] - g_q(\mathbf{y}[d])\| = \|\mathbf{n}[d]\|$  is a  $\chi^2$  random variable with mean  $\frac{n}{\text{SNR}}$  and variance  $\frac{2n}{\text{SNR}^2}$ . From (59), it is easily seen that Assumption 4 is satisfied with  $d_{\min}^2 > 0$  and  $d_{\max}^2 < \infty$ .

### E. Proof of Theorem 3

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

$$\tilde{\mathbf{u}}[m + \nu - 1] \neq \mathbf{u}[m + \nu - 1] \quad (60)$$

Theorem 3 holds if we can prove that any  $\tilde{\mathbf{u}}(D)$  satisfying (60) cannot be the ML state sequence.

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

$$\begin{aligned} K_l &= \underset{k>0}{\operatorname{argmin}}\{\tilde{\mathbf{u}}[m + \nu - 1 - k\nu] = \mathbf{u}[m + \nu - 1 - k\nu]\} \\ K_r &= \underset{k>0}{\operatorname{argmin}}\{\tilde{\mathbf{u}}[m + \nu - 1 + k\nu] = \mathbf{u}[m + \nu - 1 + k\nu]\}. \end{aligned} \quad (61)$$

We consider respectively the following four cases based on the values of  $K_l$  and  $K_r$ . In all the four cases, we show  $\tilde{\mathbf{u}}(D)$  cannot be the ML sequence.

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

Since  $\tilde{\mathbf{u}}[m + \nu - 1 + k\nu] \neq \mathbf{u}[m + \nu - 1 + k\nu]$  for all  $-K_l < k < K_r$ , according to Assumption 2,  $\tilde{\mathbf{y}}(D)$  and  $\mathbf{y}(D)$  differ at no less than  $\lfloor \frac{K_l + K_r}{2} \rfloor$  time indices in the time interval  $[m + \nu - K_l\nu, m + \nu + K_r\nu)$ , where  $\lfloor x \rfloor$  denotes the maximum integer no larger than  $x$ . According to (33) and (36), for  $d \in [m - 2M\nu, m + 2M\nu)$ , if  $\tilde{\mathbf{y}}[d] \neq \mathbf{y}[d]$ , we have

$$-\log \frac{f_o(\mathbf{r}[d]|\tilde{\mathbf{y}}[d])}{f_o(\mathbf{r}[d]|\mathbf{y}[d])} \geq L_l(\mathbf{r}[d], \mathbf{y}[d]) > 3\nu(\rho - \log p_{tr}). \quad (62)$$

Consequently, we get

$$\begin{aligned} & - \sum_{d=m+\nu-K_l\nu}^{m+\nu-1+K_r\nu} \log \frac{f_o(\mathbf{r}[d]|\tilde{\mathbf{y}}[d])P_t(\tilde{\mathbf{u}}[d]|\tilde{\mathbf{u}}[d-1])}{f_o(\mathbf{r}[d]|\mathbf{y}[d])P_t(\mathbf{u}[d]|\mathbf{u}[d-1])} \\ & \geq \left\lfloor \frac{K_l + K_r}{2} \right\rfloor 3\nu(\rho - \log p_{tr}) + (K_r + K_l)\nu \log p_{tr} \geq \left\lfloor \frac{K_l + K_r}{2} \right\rfloor 3\nu\rho > 0 \end{aligned} \quad (63)$$

According to the PCC presented in Appendix A, (63) implies that  $\mathbf{u}(D)$  “covers”<sup>7</sup>  $\tilde{\mathbf{u}}(D)$ . Hence  $\tilde{\mathbf{u}}(D)$  cannot be the ML sequence.

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

In this case, we will construct a Markov sequence  $\mathbf{u}_c(D)$  and show that  $\mathbf{u}_c(D)$  covers  $\tilde{\mathbf{u}}(D)$ .

<sup>7</sup>See definition in Appendix A.

$\mathbf{u}_c(D)$  is constructed as follows.

$$\begin{aligned}\mathbf{u}_c[d] &= \mathbf{u}[d], & \text{for } d < m + 2M\nu \\ \mathbf{u}_c[d] &= \tilde{\mathbf{u}}[d], & \text{for } d \geq m + (2M + 1)\nu.\end{aligned}\quad (64)$$

According to (31), we can always construct  $\mathbf{u}_c[d]$  for  $d \in [m + 2M\nu, m + (2M + 1)\nu)$  so that (64) is satisfied. Let  $\mathbf{y}_c(D)$  be the processed state sequence corresponding to  $\mathbf{u}_c(D)$ .

From (33) and the first inequality in (37), we get

$$\begin{aligned}& - \sum_{d=m+2M\nu}^{m+(2M+1)\nu} \log \frac{f_o(\mathbf{r}[d]|\tilde{\mathbf{y}}[d])P_t(\tilde{\mathbf{u}}[d]|\tilde{\mathbf{u}}[d-1])}{f_o(\mathbf{r}[d]|\mathbf{y}_c[d])P_t(\mathbf{u}_c[d]|\mathbf{u}_c[d-1])} \\ & \geq - \sum_{d=m+2M\nu}^{m+(2M+1)\nu-1} L_u(\mathbf{r}[d], \mathbf{y}[d]) + (\nu + 1) \log p_{tr} \geq -3M\nu\rho\end{aligned}\quad (65)$$

Since  $\tilde{\mathbf{u}}[m + \nu - 1 + k\nu] \neq \mathbf{u}_c[m + \nu - 1 + k\nu]$  for all  $-K_l < k \leq 2M - 1$ , according to Assumption 2,  $\tilde{\mathbf{y}}(D)$  and  $\mathbf{y}(D)$  differ at no less than  $\lfloor \frac{K_l + 2M - 1}{2} \rfloor$  time indices in the time interval  $[m + \nu - K_l\nu, m + 2M\nu)$ . According to (33) and (36), we have

$$\begin{aligned}& - \sum_{d=m+\nu-K_l\nu}^{m+2M\nu-1} \log \frac{f_o(\mathbf{r}[d]|\tilde{\mathbf{y}}[d])P_t(\tilde{\mathbf{u}}[d]|\tilde{\mathbf{u}}[d-1])}{f_o(\mathbf{r}[d]|\mathbf{y}_c[d])P_t(\mathbf{u}_c[d]|\mathbf{u}_c[d-1])} \\ & > \left\lfloor \frac{K_l + 2M - 1}{2} \right\rfloor 3\nu(\rho - \log p_{tr}) + (K_l + 2M - 1)\nu \log p_{tr} \\ & \geq 3M\nu(\rho - \log p_{tr}) + 2M\nu \log p_{tr} \geq 3M\nu\rho\end{aligned}\quad (66)$$

Combining (65) and (66), we obtain

$$- \sum_{d=m+\nu-K_l\nu}^{m+(2M+1)\nu} \log \frac{f_o(\mathbf{r}[d]|\tilde{\mathbf{y}}[d])P_t(\tilde{\mathbf{u}}[d]|\tilde{\mathbf{u}}[d-1])}{f_o(\mathbf{r}[d]|\mathbf{y}_c[d])P_t(\mathbf{u}_c[d]|\mathbf{u}_c[d-1])} > 0 \quad (67)$$

(67) implies that  $\mathbf{u}_c(D)$  covers  $\tilde{\mathbf{u}}(D)$ . Hence according to the PCC,  $\tilde{\mathbf{u}}(D)$  cannot be the ML sequence.

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

Similar to Case 2, we will construct a Markov sequence  $\mathbf{u}_c(D)$  and show that  $\mathbf{u}_c(D)$  covers  $\tilde{\mathbf{u}}(D)$ .

$\mathbf{u}_c(D)$  is constructed as follows.

$$\begin{aligned}\mathbf{u}_c[d] &= \mathbf{u}[d], & \text{for } d \geq m - 2M\nu \\ \mathbf{u}_c[d] &= \tilde{\mathbf{u}}[d], & \text{for } d < m - (2M + 1)\nu.\end{aligned}\quad (68)$$

According to (31), we can always construct  $\mathbf{u}_c[d]$  for  $d \in [m - (2M + 1)\nu, m - 2M\nu)$  so that (68) is satisfied. Let  $\mathbf{y}_c(D)$  be the processed state sequence corresponding to  $\mathbf{u}_c(D)$ .

From (33) and the second inequality in (37), we get

$$\begin{aligned}& - \sum_{d=m-(2M+1)\nu}^{m-2M\nu-1} \log \frac{f_o(\mathbf{r}[d]|\tilde{\mathbf{y}}[d])P_t(\tilde{\mathbf{u}}[d]|\tilde{\mathbf{u}}[d-1])}{f_o(\mathbf{r}[d]|\mathbf{y}_c[d])P_t(\mathbf{u}_c[d]|\mathbf{u}_c[d-1])} \\ & \geq - \sum_{d=m-(2M+1)\nu}^{m-2M\nu-1} L_u(\mathbf{r}[d], \mathbf{y}[d]) + \nu \log p_{tr} \geq -3M\nu\rho.\end{aligned}\quad (69)$$

Since  $\tilde{\mathbf{u}}[m + \nu - 1 + k\nu] \neq \mathbf{u}_c[m + \nu - 1 + k\nu]$  for all  $-2M - 1 \leq k < K_r$ , according to Assumption 2,  $\tilde{\mathbf{y}}(D)$  and  $\mathbf{y}(D)$  differ at no less than  $\left\lfloor \frac{2M+1+K_r}{2} \right\rfloor$  time indices in the time interval  $[m - 2M\nu, m + \nu + K_r\nu]$ . According to (33) and (36), we have

$$\begin{aligned} & - \sum_{d=m-2M\nu}^{m+\nu+K_r\nu-1} \log \frac{f_o(\mathbf{r}[d]|\tilde{\mathbf{y}}[d])P_t(\tilde{\mathbf{u}}[d]|\tilde{\mathbf{u}}[d-1])}{f_o(\mathbf{r}[d]|\mathbf{y}_c[d])P_t(\mathbf{u}_c[d]|\mathbf{u}_c[d-1])} \\ & > \left\lfloor \frac{2M+1+K_r}{2} \right\rfloor 3\nu(\rho - \log p_{tr}) + (2M+1+K_r)\nu \log p_{tr} \geq 3(M+1)\nu\rho. \end{aligned} \quad (70)$$

Combining (69) and (70), we obtain

$$\sum_{d=m-(2M+1)\nu}^{m+\nu+K_r\nu-1} \log \frac{f_o(\mathbf{r}[d]|\tilde{\mathbf{y}}[d])P_t(\tilde{\mathbf{u}}[d]|\tilde{\mathbf{u}}[d-1])}{f_o(\mathbf{r}[d]|\mathbf{y}_c[d])P_t(\mathbf{u}_c[d]|\mathbf{u}_c[d-1])} < 0 \quad (71)$$

(71) implies that  $\mathbf{u}_c(D)$  covers  $\tilde{\mathbf{u}}(D)$ . Hence according to the PCC,  $\tilde{\mathbf{u}}(D)$  cannot be the ML sequence.

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

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

$$\begin{aligned} \mathbf{u}_c[d] &= \mathbf{u}[d], & \text{for } m - 2M\nu \leq d < m + 2M\nu \\ \mathbf{u}_c[d] &= \tilde{\mathbf{u}}[d], & \text{for } d \geq m + (2M + 1)\nu \\ \mathbf{u}_c[d] &= \tilde{\mathbf{u}}[d], & \text{for } d < m - (2M + 1)\nu. \end{aligned} \quad (72)$$

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

Since  $\tilde{\mathbf{u}}[m + \nu - 1 + k\nu] \neq \mathbf{u}_c[m + \nu - 1 + k\nu]$  for all  $-2M - 1 \leq k \leq 2M - 1$ , according to Assumption 2,  $\tilde{\mathbf{y}}(D)$  and  $\mathbf{y}(D)$  differ at no less than  $\left\lfloor \frac{4M+1}{2} \right\rfloor$  time indices in the time interval  $[m - 2M\nu, m + 2M\nu]$ . According to (33) and (36), we have

$$- \sum_{d=m-2M\nu}^{m+2M\nu-1} \log \frac{f_o(\mathbf{r}[d]|\tilde{\mathbf{y}}[d])P_t(\tilde{\mathbf{u}}[d]|\tilde{\mathbf{u}}[d-1])}{f_o(\mathbf{r}[d]|\mathbf{y}_c[d])P_t(\mathbf{u}_c[d]|\mathbf{u}_c[d-1])} > \left\lfloor \frac{4M+1}{2} \right\rfloor 3\nu(\rho - \log p_{tr}) + 4M\nu \log p_{tr} \geq 6M\nu\rho. \quad (73)$$

Meanwhile, it is easily seen that (65) and (69) hold. Combine (65), (69) and (73), we obtain

$$- \sum_{d=m-(2M+1)\nu}^{m+(2M+1)\nu} \log \frac{f_o(\mathbf{r}[d]|\tilde{\mathbf{y}}[d])P_t(\tilde{\mathbf{u}}[d]|\tilde{\mathbf{u}}[d-1])}{f_o(\mathbf{r}[d]|\mathbf{y}_c[d])P_t(\mathbf{u}_c[d]|\mathbf{u}_c[d-1])} > -3M\nu\rho - 3M\nu\rho + 6M\nu\rho = 0. \quad (74)$$

(74) implies that  $\mathbf{u}_c(D)$  covers  $\tilde{\mathbf{u}}(D)$ . Hence according to the PCC,  $\tilde{\mathbf{u}}(D)$  cannot be the ML sequence.

Overall, we showed that  $\tilde{\mathbf{u}}(D)$  cannot be the ML sequence irrespective of the values of  $K_l$  and  $K_r$ . Therefore,  $\tilde{\mathbf{u}}[m + \nu - 1] = \mathbf{u}[m + \nu - 1]$  must be true.  $\blacksquare$

## REFERENCES

- [1] G. Forney, *The Viterbi Algorithm*, Proc. of The IEEE, Vol. 61, No. 3, pp. 268-278, Mar. 1973.
- [2] A. Viterbi, *Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm*, IEEE Trans. Inform. Theory, Vol. IT-13, No. 2, pp. 260-269, Apr. 1967.

- [3] K. Zigangirov and H. Osthoff, *List Decoding of Trellis Codes*, Problems of Control and Information Theory, pp. 347-364, 1980.
- [4] R. Fano, *A Heuristic Discussion of Probabilistic Decoding*, IEEE Trans. Inform. Theory, Vol. IT-9, pp. 64-74, Apr. 1963.
- [5] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, *Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate*, IEEE Trans. Inform. Theory, Vol. IT-20, pp. 284-287, Mar. 1974.
- [6] R. Johannesson and K. Zigangirov, *Fundamentals of Convolutional Coding*, IEEE Press, 1999.
- [7] P. Swaszek and W. Jones, *How Often Is Hard-Decision Decoding Enough?*, IEEE Trans. Inform. Theory, Vol. 44, pp. 1187-1193, May 1998.
- [8] M. Ariel and J. Snyders, *Error-Trellises for Convolutional Codes-Part II: Decoding Methods*, IEEE Trans. Commun., Vol. 47, pp. 1015-1024, July 1999.
- [9] B. Hassibi and H. Vikalo, *On The Sphere Decoding Algorithm I. Expected Complexity*, IEEE Trans. Sig. Proc., Vol. 53, No. 8, pp. 2806-2818, Aug. 2005.
- [10] 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.
- [11] H. Vikalo and B. Hassibi, *Maximum-Likelihood Sequence Detection of Multiple Antenna Systems over Dispersive Channels via Sphere Decoding*, EURASIP J. Appl. Sig. Proc., No. 1, pp. 525-531, Jan. 2002.
- [12] H. Vikalo, *Sphere Decoding Algorithms for Digital Communications*, Ph.D. Thesis, Stanford Univ., 2003.
- [13] L. Rabiner, *A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition*, Proc. of IEEE, Vol. 77, No. 2, pp. 257-286, Feb. 1989.
- [14] G. Forney, *Structural Analysis of Convolutional Codes via Dual Codes*, IEEE Trans. Inform. Theory, Vol. IT-19, pp. 512-518, Jul. 1973.
- [15] J. Luo, *Fast Maximum Likelihood Sequence Detection over Vector Intersymbol Interference Channels*, IEEE ICASSP, Honolulu, Hawaii, Apr. 2007.