SPHERE-CONSTRAINED ML DETECTION FOR FREQUENCY-SELECTIVE CHANNELS

H. Vikalo, B. Hassibi

Department of Electrical Engineering
California Institute of Technology
Pasadena, CA 91125

U. Mitra

Department of Electrical Engineering
University of Southern California
Los Angeles, CA 90089

ABSTRACT

Maximum-likelihood (ML) detection problem for channels with memory is investigated. The Viterbi algorithm provides an elegant solution, but is computationally inefficient when employed for detection on long channels. On the other hand, sphere decoding solves the ML detection problem in polynomial expected time over a wide range of SNRs. In this paper, the sphere-constrained search strategy of sphere decoding is combined with the dynamic programming principles of the Viterbi algorithm. The resulting algorithm has the worst-case complexity of the Viterbi algorithm, but significantly lower expected complexity.

1. INTRODUCTION

We consider the frequency-selective channel model, with input/output relation given by

$$x_i = \sum_{j=1}^{l} h_j s_{i-j} + v_i,$$

where $h_i, i = 1, \ldots, l$ are the coefficients of the channel impulse response, l denotes the channel length, s_i is the i^{th} symbol in the transmitted sequence, and v_i denotes a Gaussian noise sample $\mathcal{N}(0, \sigma^2)$. The maximum-likelihood sequence detector solves the optimization problem

$$\min \sum_{m=1}^{k} |x_m - \sum_{j=1}^{l} h_j s_{m-j+1}|^2, \qquad (1)$$

to find the most likely transmitted symbol sequence. The Viterbi algorithm ([1], [2]) solves problem (1) using dynamic programming ideas [3]. However, the computational complexity of the Viterbi algorithm is exponential in the length of the channel. On the other hand, the sphere decoding algorithm [4] can also be used for ML detection on channels with memory [5], [6] – assuming a finite length for the transmitted symbol sequence. With a probabilistic choice of the search radius [7], the computational complexity of the sphere decoding algorithm is a random variable, with the mean often significantly below the complexity of the Viterbi algorithm. Therefore, there is merit in studying the possibility of combining the benefits of both algorithms. To establish the connection between the two detection techniques, it will be beneficial to first review the basic ideas of the Viterbi and the sphere decoding algorithms.

2. ML DETECTION WITH VITERBI ALGORITHM

The Viterbi algorithm is commonly defined on a trellis, a directed graph that describes systems with memory. An example of the trellis is shown in Figure 1.

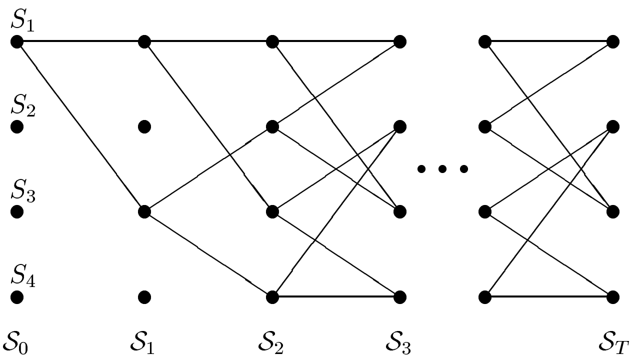

Fig. 1. *Trellis example*

The vectors of vertical black dots in the trellis in Figure 1 denote the state sets $\mathcal{S}$. These vectors are labeled by $k = 1, \ldots, T$, and arranged into an array of length T. The size of the state set $\mathcal{S}_k$ depends on the channel memory. In particular, each element of the set $\mathcal{S}_k$ represent one possible state of the channel memory. Adjacent state sets in the trellis are connected via branches. The branches are typically labeled to describe the input/output relation of the system corresponding to the particular state transition. There is a total of L branches emanating from each state, corresponding to the L possible values of the input. [Note that the trellis in Figure 1 starts from the all zeros state.] The state that a branch sinks in to depends upon the source state and the value of the current input bit. A concatenated sequence of trellis branches is called a *path*. The length of the path is determined by the number of branches in it. Each path corresponds to a distinct sequence of input symbols, indicated by the labels on the branches that comprise the path.

The Viterbi algorithm uses the trellis representation to find the solution to (1). In particular, it finds the trellis path corresponding to the smallest metric in (1) without performing an exhaustive search over the entire trellis. To this end, we describe the argument of the ML optimization problem by

$$\mathcal{C}_{k+1} = \sum_{j=1}^{k+1} |x_j - \sum_{m=1}^{l} h_m s_{j-m+1}|^2. \tag{2}$$

The value of $\mathcal{C}_{k+1}$ depends on the current state and the trellis path that led to that particular state. To find the ML optimal sequence, one needs to determine the trellis path with the smallest cost. An exhaustive search over all possible paths would clearly not be feasible even for trellises with moderate number of states. However, $\mathcal{C}_{k+1}$ can be expressed as

$$\mathcal{C}_{k+1} = \mathcal{C}_k + |x_{k+1} - \sum_{j=1}^{l} h_j s_{k-j+1}|^2. \tag{3}$$

Clearly, the second term of the RHS of (3) does not depend on $s_{k-l}, \ldots, s_0$. Therefore, to find the smallest cost path to the state S_j in the set $\mathcal{S}_{k+1}$, it is sufficient to consider all possible state transitions to S_j (along the L branches emanating from the states in set $\mathcal{S}_k$). This procedure can be done recursively. Finally, the trellis path of length T that has the smallest cost $\mathcal{C}_T$ is the optimal path. The signal sequence that corresponds to the branch transitions along the optimal trellis path is the solution to the ML detection problem.

The complexity of the Viterbi algorithm is proportional to the number of states and thus grows exponentially with the length of the channel. On the other hand, it is linear in the length of the data sequence.

3. ML DETECTION WITH SPHERE DECODING

Another technique that provides the solution to the ML detection problem without actually performing an exhaustive search is sphere decoding. To employ sphere decoding, we need to write the channel model as

$$x = Hs + v, \tag{4}$$

where $s = [s_1 \ s_2 \ \ldots \ s_T]'$ is the vector of transmitted data sequence, and $v = [v_1 \ v_2 \ \ldots \ v_{T+l-1}]'$ is the vector of additive white Gaussian noise. Matrix $H \in \mathcal{R}^{(T+l-1)\times T}$ is given by

$$H = \begin{bmatrix} h_1 & & & & & \\ h_2 & h_1 & & & & \\ \vdots & \vdots & \ddots & & & \\ h_l & \ldots & & h_1 & & \\ & \ddots & & & \ddots & \\ & & h_l & \ldots & & h_1 \\ & & & \ddots & \vdots & \vdots \\ & & & & h_l & h_{l-1} \\ & & & & & h_l \end{bmatrix}, \ h_i \in \mathcal{R}.$$

ML detection can now be expressed as

$$\min_s \|x - Hs\|^2. \tag{5}$$

This problem has a geometric interpretation: given a point x, find the closest lattice point in a skewed lattice Hs. The sphere decoding algorithm solves (5) by performing a search over only those points Hs that belong to a sphere around x. The radius r of the sphere is chosen so that we find a point inside the sphere with a high probability. In particular,

$$\|x - Hs\|^2 = \|v\|^2 = v_1^2 + \cdots + v_T^2,$$

is a chi-square random variable with T degrees of freedom. Thus the radius $r^2 = \alpha T \sigma^2$ can be chosen so that

$$\int_0^{\alpha T} \frac{\lambda^{T-1}}{\Gamma(T)} e^{-\lambda} d\lambda = p_{\text{fp}}, \tag{6}$$

where $p_{\text{fp}} = 0.99$, say.

The condition that point Hs belongs to the sphere of radius r is given by

$$r^2 \geq \|x - Hs\|^2. \tag{7}$$

The RHS of (7) can be expanded as

$$\begin{aligned} r^2 \ &\geq \ (x_1 - h_1 s_1)^2 \\ &+ \ (x_2 - h_1 s_2 - h_2 s_1)^2 + \ldots \end{aligned} \tag{8}$$

where the first term depends only on s_1, the second term on $\{s_1, s_2\}$ and so on. Therefore, considering the first term only, a necessary condition for Hs to lie inside the hypersphere is

$$r^2 \geq (x_1 - h_1 s_1)^2. \tag{9}$$

This condition is equivalent to s_1 belonging to the interval

$$\left\lceil \frac{-r + x_1}{h_1} \right\rceil \leq s_1 \leq \left\lfloor \frac{r + x_1}{h_1} \right\rfloor. \tag{10}$$

Furthermore, for every s_1 satisfying (10), s_2 needs to satisfy

$$r^2 \geq (x_1 - h_1 s_1)^2 + (x_2 - h_1 s_2 - h_2 s_1)^2 \tag{11}$$

Defining

$$r_2^2 = r^2 - (x_1 - h_1 s_1)^2 \,, \tag{12}$$

and $x_{2|1} = x_2 - h_2 s_1$, a stronger necessary condition can be found by looking at the first two terms in (8), which leads to s_2 belonging to the interval

$$\left\lceil \frac{-r_2 + x_{2|1}}{h_1} \right\rceil \leq s_2 \leq \left\lfloor \frac{r_2 + x_{2|1}}{h_1} \right\rfloor. \tag{13}$$

One can continue in a similar fashion for s_3, and so on until s_T. Note that these T conditions used to find s are necessary but still not sufficient. Only if an additional constraint,

$$\begin{aligned} r_{T+1}^2 \quad &\geq \quad (x_{T+1} - h_l s_{T-l+2} - \cdots - h_2 s_T)^2 + \ldots \\ &+ \quad (x_{T+l-1} - h_l s_T)^2, \end{aligned}$$

is satisfied, the point s indeed does belong to the sphere, i.e., it satisfies condition (7).

The sphere decoding algorithm performs a search on the tree, as illustrated in Figure 2. The nodes on the k^{th} level of the tree correspond to the vectors $[s_1 \ldots s_k]'$. Since the vector x in (4) is not arbitrary, but is a point Hs perturbed by additive noise with known statistical properties, we can talk about the average complexity of the sphere decoding algorithm. The expected complexity of the algorithm is proportional to the expected number of points in the tree that the algorithm visits [5]. For moderate data-block lengths, it is polynomial over a wide range of SNRs.

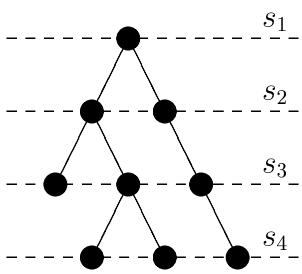

Fig. 2. *Tree search of the sphere decoding algorithm.*

4. COMBINED SPHERE DECODING AND VITERBI ALGORITHM

The Viterbi algorithm efficiently solves ML detection problem on trellises with a moderate number of states. However, for long channels and/or modulations with high cardinality constellations, the Viterbi algorithm is often inefficient and occasionally non-feasible. On the other hand, the sphere decoding algorithm has expected complexity often significantly below the complexity of the Viterbi algorithm. However, the worst-case complexity of the sphere decoding algorithm is exponential and corresponds to the exhaustive search. Therefore, a hybrid receiver structure that combines the sphere decoding constrained search strategy with the trellis based decoding of the Viterbi algorithm, is desired. This can be obtained by either modifying the sphere decoding algorithm to impose the channel memory state constraints or imposing a sphere-constrained search onto the Viterbi algorithm. We briefly discuss both.

Consider the sphere decoding algorithm and the search illustrated in Figure 2. The sphere decoding algorithm does not account for the special structure (banded Toeplitz) of the lattice generating matrix in (4). We propose the following modification: Assume that the algorithm is currently examining a point on the k^{th} level of the tree. Based on the current, and up to $l-2$ points on levels $k-1, k-2, \ldots$, identify the corresponding state S_j, $j = 1, 2, \ldots, L^{l-1}$ (where the state is defined as on the trellis). Furthermore, from (12), it is easy to see (by writing out the recursion) that the cost associated with this state is given by

$$\mathcal{C}_k(S_j) = r^2 - r_{k+1}^2.$$

Now, in addition to the standard steps that the sphere decoding algorithm undertakes, it also compares this $\mathcal{C}_k(S_j)$ with the previously stored $\mathcal{C}_k(S_j)$ and, if the current one is greater, prunes the tree (i.e., discard the current tree node). If the current $\mathcal{C}_k(S_j)$ is smaller than the previously stored $\mathcal{C}_k(S_j)$ (or there are no previously stored $\mathcal{C}_k(S_j)$), it stores the current value of $\mathcal{C}_k(S_j)$ and proceeds with the other sphere decoding steps.

Alternatively, we modify the Viterbi algorithm by imposing the sphere-constrained trellis search. Consider the trellis representation of a frequency-selective channel and a finite data block transmition. We impose the constraint (7) that the transmitted signal belongs to a sphere of radius r implicitly defined by (6). As we have shown in the previous section, an obvious necessary condition that the transmitted signal needs to satisfy is given by (9). However, from (2), this condition is equivalent to the constraint $r^2 \geq \mathcal{C}_1$. Similarly, comparing (11) and (2), we obtain that $r^2 \geq \mathcal{C}_2$. In general,

$$r^2 \geq \mathcal{C}_k, \quad k = 1, 2, \ldots, T. \tag{14}$$

On the trellis, condition (14) means that the cost $\mathcal{C}_k$ should, for each state and time index k, be smaller than the radius of the sphere. The states that violate condition (14) can be neglected, i.e., no branches emanating from such states need to be considered when searching for the optimal trellis path. Therefore, the search on trellis can, on average, be performed faster that the Viterbi algorithm. The worst case complexity, on the other hand, coincides with the complexity of the Viterbi algorithm.

The sphere-constrained trellis search is illustrated in Figure 3, where the "empty" states denote those from which no branch on optimal path can emanate.

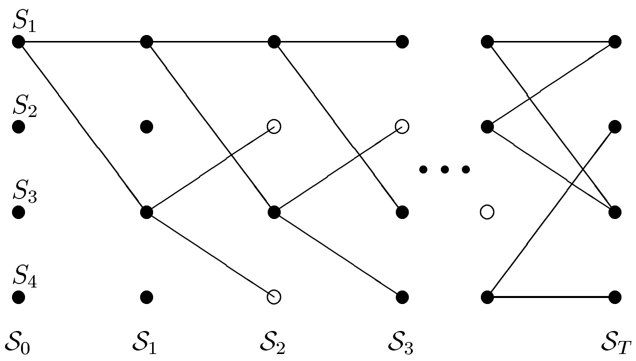

Fig. 3. *Sphere-constrained search on trellis*

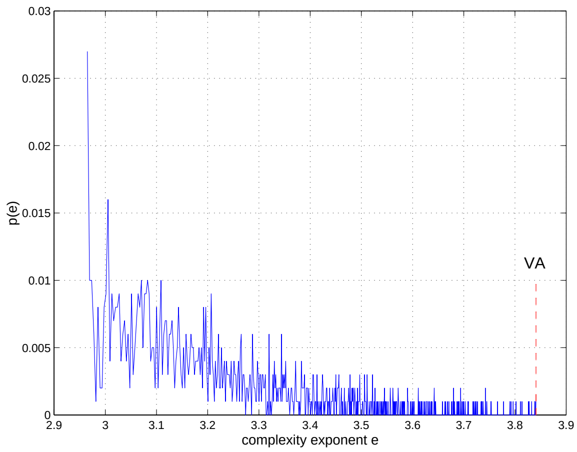

Fig. 4. *Distribution of complexity exponent,* $l = 8$, $T = 20$, $L = 2$, $SNR = 10$*dB.*

5. SIMULATION RESULTS AND SUMMARY

We consider a channel of length $l = 8$, transmitting BPSK modulated ($L = 2$) data in blocks of length $T = 20$ at $SNR = 10$dB and employ the sphere-constrained detection on the trellis. Figure 4 shows the empirical distribution of the complexity exponent, defined as

$$e = \log_T F,$$

where F denotes the total number of operations (flop count) performed when detecting s.

As evident from Figure 4, the complexity exponent is always smaller than the complexity exponent corresponding to the Viterbi algorithm (denoted by the vertical dashed line). Figure 5 shows the expected complexity exponent as a function of SNR. The expected complexity is roughly cubic in the considered SNR range.

In summary, we proposed combinations of the sphere decoding and the Viterbi algorithms for ML detection for

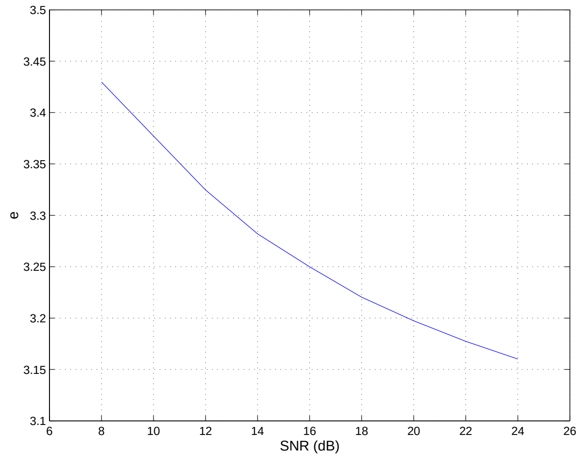

Fig. 5. *Expected complexity exponent,* $l = 8$, $T = 20$, $L = 2$.

channels with memory. The hybrid algorithm is either the sphere decoding modified to speed-up the search for the closest-point in the lattice or the Viterbi algorithm with the sphere-constrained search on the trellis. Example illustrates improvement in the computational complexity.

6. REFERENCES

[1] A. D. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithms," *IEEE Trans. on Inf. Theory*, vol. 13, pp. 260-269, 1967.

[2] G. D. Forney, "Maximum-likelihood sequence estimation of digital sequences in the presence of intersymbol interference," *IEEE Trans. of Inf. Theory*, vol. 18, no. 3, pp. 363-378, May 1972.

[3] R. Bellman, *Dynamic Programming*, Princeton University Press, Princeton, 1957.

[4] U. Fincke and M. Pohst, "Improved methods for calculating vectors of short length in a lattice, including a complexity analysis," *Mathematics of Computation*, vol. 44, pp. 463-471, April 1985.

[5] H. Vikalo and B. Hassibi, "Sphere decoding for frequency-selective channels," *Proc. 40th Allerton Conf. on Comm. Contr. and Comp.*, Monticello, 2002.

[6] M. O. Damen and, K. Abed-Meraim, and A. Safavi, "On CDMA with space-time codes over multi-path fading channels", to appear in *IEEE Trans. on Wireless*, January 2003.

[7] B. Hassibi and H. Vikalo, "Expected complexity of the sphere decoder algorithm," *submitted to the IEEE Trans. on Signal Processing*, 2002.