# Sphere-Constrained ML Detection for Frequency-Selective Channels

H. Vikalo, B. Hassibi, and U. Mitra, *Senior Member, IEEE*

*Abstract*—The maximum-likelihood (ML) sequence detection problem for channels with memory is investigated. The Viterbi algorithm (VA) provides an exact solution. Its computational complexity is linear in the length of the transmitted sequence, but exponential in the channel memory length. On the other hand, the sphere decoding (SD) algorithm also solves the ML detection problem exactly, and has expected complexity which is a low-degree polynomial (often cubic) in the length of the transmitted sequence over a wide range of signal-to-noise ratios. We combine the sphere-constrained search strategy of SD with the dynamic programming principles of the VA. The resulting algorithm has the worst-case complexity determined by the VA, but often significantly lower expected complexity.

*Index Terms*—Expected complexity, frequency-selective channels, maximum-likelihood (ML) sequence detection, sphere decoding, Viterbi algorithm (VA).

## I. INTRODUCTION

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

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

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 chosen from an $L$-pulse amplitude modulation (PAM) constellation $\mathcal{D}_L = \{-(L-1/2), \ldots, (L-1/2)\}$, and $v_i$ denotes a Gaussian noise sample $\mathcal{N}(0, \sigma^2)$. Over a horizon of length $T$, the maximum-likelihood (ML) sequence detector minimizes the cost function

$$\mathcal{C}_T = \sum_{k=1}^{T} \left| x_k - \sum_{m=1}^{l} h_m s_{k-m+1} \right|^2 \qquad (2)$$

to find the most likely transmitted symbol sequence $\{s_1, \ldots, s_T\}$. The Viterbi algorithm (VA) [1] exploits the Markovian property of the channel, and finds the sequence which minimizes $\mathcal{C}_T$ by using dynamic programming ideas. Typically, the VA is employed as a breadth-first search on a trellis, a directed graph describing systems with memory, illustrated in Fig. 1. The key observation is that $\mathcal{C}_T$ can be

H. Vikalo and B. Hassibi are with the Department of Electrical Engineering, California Institute of Technology, Pasadena, CA 91125 USA (e-mail: hvikalo@systems.caltech.edu).

U. Mitra is with the Department of Electrical Engineering, University of Southern California, Los Angeles, CA 90089 USA.
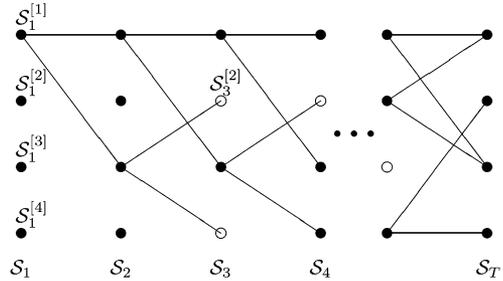
Fig. 1. Trellis with four states. The "empty" nodes indicate states which violate the sphere constraint and are thus pruned by the combined VA/SD. For instance, as indicated above, $\mathcal{C}_3(\mathcal{S}_3^{[2]}) > r^2$.

recursively computed as

$$\mathcal{C}_{k+1} = \mathcal{C}_k + \left| x_{k+1} - \sum_{m=1}^{l} h_m s_{k-m+1} \right|^2,$$
$$k = 0, \ldots, T-1, \mathcal{C}_0 = 0. \qquad (3)$$

Clearly, the second term on the right-hand side (RHS) of (3) does not depend on $s_{k-l}, \ldots, s_1$, but only on the current symbol $s_k$ and the current memory of the channel $s_{k-1}s_{k-2}\cdots s_{k-l+1}$. The $L^{l-1}$ possible states of the channel memory comprise the state set $\mathcal{S}_k$ (see Fig. 1). To find the $k$th segment of the optimal path, it is sufficient, for every state $\mathcal{S}_{k+1}^{[j]} \in \mathcal{S}_{k+1}$ ($j = 1, \ldots, L^{l-1}$), to keep only the branch emanating from a state in $\mathcal{S}_k$ and terminating in the state $\mathcal{S}_{k+1}^{[j]}$ that has the smallest transition cost. This procedure can be done recursively. 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 VA 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.

The sphere decoding (SD) algorithm [2] can also be used for detection on channels with memory [3], [4]. Permitting a guard interval, the SD algorithm provides ML performance. To employ SD, we first need to write (1) as

$$\mathbf{x} = H\mathbf{s} + \mathbf{v}$$

where the $(T+l-1) \times 1$ vectors $\mathbf{x}$ and $\mathbf{v}$, the $(T+l-1) \times T$ matrix $H$, and the $T \times 1$ vector $\mathbf{s}$ are given by

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{T+l-1} \end{bmatrix}, \quad \mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_{T+l-1} \end{bmatrix}$$

$$H = \begin{bmatrix} h_1 \\ \vdots & \ddots \\ h_l & \cdots & h_1 \\ & \ddots & \\ & & h_l \end{bmatrix}, \quad \mathbf{s} = \begin{bmatrix} s_1 \\ \vdots \\ s_T \end{bmatrix}. \qquad (4)$$

The ML detection can now be expressed as an integer least-squares problem

$$\min_{\mathbf{s}\in\mathcal{D}_L^T} \|\mathbf{x} - H\mathbf{s}\|^2. \tag{5}$$

This problem has a geometric interpretation: given a point $\mathbf{x}$, find the closest lattice point in a skewed lattice $H\mathbf{s}$. The SD algorithm solves (5) by performing a search over only those points $H\mathbf{s}$ that belong to a sphere around $\mathbf{x}$. The radius $r$ of the sphere is chosen so that we find a point inside the sphere with a high probability. We assume that the noise samples $v_1, v_2, \ldots, v_T$ are independent and identically distributed (i.i.d.); hence $\|\mathbf{x} - H\mathbf{s}\|^2 = \|\mathbf{v}\|^2 = v_1^2 + \cdots + v_T^2$ is a chi-square random variable with $T$ degrees of freedom. Thus, the radius $r^2 = \alpha T \sigma^2$ can be chosen probabilistically so that

$$\int_0^{\alpha T} \frac{\lambda^{T/2-1}}{\Gamma(T)} e^{-\lambda} d\lambda = 1 - \epsilon \tag{6}$$

where $\epsilon \ll 1$. The condition that a point $H\mathbf{s}$ belongs to the sphere of radius $r$ is given by

$$r^2 \geq \|\mathbf{x} - H\mathbf{s}\|^2. \tag{7}$$

The summation on the RHS of (7) can be expanded to yield a set of conditions on the components of $\mathbf{s}$

$$(x_1 - h_1 s_1)^2 \leq r_1^2, (x_2 - h_1 s_2 - h_2 s_1)^2 \leq r_2^2,$$
$$(x_3 - h_1 s_3 - h_2 s_2 - h_3 s_1)^2 \leq r_3^2, \text{ etc.}$$

where $r_1^2 = r^2$, $r_2^2 = r_1^2 - (x_1 - h_1 s_1)^2$, $r_3^2 = r_2^2 - (x_2 - h_1 s_2 - h_2 s_1)^2$, etc. Note that this gives $T$ conditions on the components of $\mathbf{s}$ which are necessary, but still not sufficient. Only if the additional constraint

$$r_{T+1}^2 \geq (x_{T+1} - h_l s_{T-l+2} - \cdots - h_2 s_T)^2 + \ldots$$
$$+ (x_{T+l-1} - h_l s_T)^2$$

where $r_{T+1}^2 = r^2 - \sum_{j=1}^T (x_j - \sum_{m=1}^l h_m s_{j-m+1})^2$ is satisfied, does the point $\mathbf{s}$ indeed belong to the sphere, i.e., satisfies condition (7). The SD algorithm performs a depth-first search on a tree, as illustrated in Fig. 2. A trace leading to a surviving node on the $k$th level of the tree corresponds to a vector $[s_1 \cdots s_k]^*$ inside the $k$-dimensional sphere. With the probabilistic choice of $r$, the computational complexity of the SD algorithm is a random variable [5], with the mean often significantly below the complexity of the VA [4].

## II. COMBINING SD AND VA

The complexity of the VA is linear in the length of the data sequence, but is exponential in the channel memory size, where the base of the exponent is the symbol alphabet size. Thus, for long channels and/or large symbol alphabets, the VA is often inefficient and occasionally nonfeasible. On the other hand, over a wide range of signal-to-noise ratios (SNRs), the expected complexity of SD is a low-degree polynomial in the data block length, and the degree of the polynomial does not
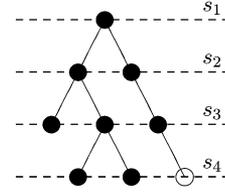


Fig. 2. Tree-search interpretation of the SD algorithm. The "empty" node indicates a lattice point which is inside a 4-D sphere, but violates corresponding state constraint (i.e., $\mathcal{C}_4(\cdot)$ corresponding to this point is not the smallest one), and is thus pruned by the combined SD/VA.

vary significantly with the channel memory size. However, the worst-case complexity of SD is exponential in the data block length. Furthermore, the SD algorithm does not at all exploit the Markovian property of the channel, which is precisely what the VA does. Therefore, a hybrid receiver structure that combines the constrained search strategy of SD with the trellis-based decoding of the VA is desired. This can be obtained by either modifying the SD algorithm to include the channel memory state constraints or by adding the sphere constraints to the trellis search of the VA. The two approaches, essentially equivalent, although one is depth-first and the other is breadth-first, are briefly presented here.

*Algorithm 1 [SD Modified with VA]:* Consider the SD algorithm and the tree search illustrated in Fig. 1. The SD algorithm does not take into account the Markovian property of the channel. We propose the following modification. Assume that the algorithm is currently examining a node on the $k$th level of the tree. Based on the current, and on up to $l - 2$ tree nodes on levels $k - 1, k - 2, \ldots, k - l + 1$, along which the algorithm descended to the current node, we identify the current state $\mathcal{S}_k^{[j]} = s_k s_{k-1} \ldots s_{k-l+1}$. The meaning of the state is the same as on the trellis, i.e., it is the current state of the channel memory and we assign a cost $\mathcal{C}_k(\mathcal{S}_k^{[j]})$ to it. By writing out the recursion for $r_k^2$, it is easy to see that

$$\mathcal{C}_k\left(\mathcal{S}_k^{[j]}\right) = r^2 - r_{k+1}^2.$$

Now, in addition to the standard steps that the SD algorithm undertakes, we enforce that it also compares this $\mathcal{C}_k(\mathcal{S}_k^{[j]})$ with the previously stored minimum cost $\min \mathcal{C}_k(\mathcal{S}_k^{[j]})$. If the current $\mathcal{C}_k(\mathcal{S}_k^{[j]})$ is greater than $\min \mathcal{C}_k(\mathcal{S}_k^{[j]})$, the algorithm prunes the tree, i.e., it discards the current tree node. If the current $\mathcal{C}_k(\mathcal{S}_k^{[j]})$ is smaller than the previously stored $\min \mathcal{C}_k(\mathcal{S}_k^{[j]})$ (or there are no previously stored $\min \mathcal{C}_k(\mathcal{S}_k^{[j]})$'s), the algorithm assigns $\min \mathcal{C}_k(\mathcal{S}_k^{[j]}) := \mathcal{C}_k(\mathcal{S}_k^{[j]})$ and proceeds with the other SD steps. Note that the algorithm is still depth-first. Clearly, its complexity will be lower than the complexity of the original SD, at the expense of additional storage requirements. $\square$

*Algorithm 2 [VA Modified with SD]:* Consider the trellis representation of a frequency-selective channel and a finite data block transmission. We impose the constraint (7) that the transmitted signal belongs to a sphere of radius $r$ 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

$(x_1 - h_1 s_1)^2 \leq r_1^2$. However, from (2), this condition is equivalent to the constraint $\mathcal{C}_1(\mathcal{S}_1^{[j]}) \leq r^2$ for all $j$. Similarly, condition $(x_2 - h_1 s_2 - h_2 s_1)^2 \leq r_2^2$ is equivalent to the constraint $\mathcal{C}_2(\mathcal{S}_2^{[j]}) \leq r^2$ for all $j$. In general

$$\mathcal{C}_k(\mathcal{S}_k^{[j]}) \leq r^2, \quad k = 1, 2, \ldots, T, \quad j = 1, 2 \ldots, L^{\min(k,l)-1}. \tag{8}$$

On the trellis, condition (8) means that the cost $\mathcal{C}_k(\mathcal{S}_k^{[j]})$ should, for every state index $j$ and every time index $k$, be smaller than the radius of the sphere. The states $\mathcal{S}_k^{[j]}$ that violate condition (8) 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 than the VA. The worst-case complexity, on the other hand, coincides with the complexity of the VA. The sphere-constrained trellis search is illustrated in Fig. 1. □

The following points are worthy of mention. *Algorithm 2* is employed on the trellis, and essentially reduces the complexity of the VA by discarding the states which violate certain (sphere) constraints. Hence, *Algorithm 2* is, in fact, a reduced-state detection algorithm; in particular, it is closely related to the T-algorithm [6], which discards states whose cost is greater than some threshold. In this letter, the lattice structure of the modulation scheme is used to relate the aforementioned threshold to the radius of the sphere centered at the received vector, where this radius is chosen according to the noise variance, so it guarantees finding the solution with high probability. Such a choice of the radius/threshold enables us to analytically find the expected complexity of *Algorithm 2*, which we derive in the next section. A probabilistic choice of the threshold was also recently proposed in [7], where it was chosen based on the probability of discarding an optimal state. However, unlike the majority of state-reduced algorithms, *Algorithm 2* does not sacrifice the ML performance, but (permitting a guard interval) solves the detection problem exactly. Note that obtaining the exact solution may actually require increasing the sphere radius, if no point is found inside, and repeating the search.

Also, recall that the VA is often employed for sequence detection by forcing the detector to make decisions once sufficiently deep inside the trellis (common heuristic suggests that five times the channel length is sufficiently deep). *Algorithm 2* can be slightly modified to employ the same heuristic. In this case, one can think of a sliding window (or a "sliding sphere") of length (dimension) $5l$ that imposes a sphere constraint of the form (7) on the states of the trellis.

On another note, *Algorithm 1* is the more preferable one when the ratio $l/T$ is relatively large; the reason is its depth-first strategy, which allows one to, when the algorithm finds a point inside the sphere, compute a new radius based on that point and restart, thus making the search for the optimal solution more efficient. It also allows for implementation of computationally more efficient versions of sphere constrained search, such as the Schnorr–Euchner (see [4], [5], and the references therein).

## III. THE EXPECTED COMPLEXITY OF THE COMBINED VA/SD

In this section, we find an analytic expression for the expected complexity of the combined VA/SD proposed above. The

derivation takes the approach originally proposed in [4] and [5], where it was used to find the expected complexity of the original SD algorithm of Fincke and Pohst [2]. Clearly, the expected complexity of the combined VA/SD is proportional to the expected number of the states that survive the pruning process

$$E_c = \sum_{d=1}^{T} \underbrace{E(\text{\# of states that survive in the state set } \mathcal{S}_d)}_{E_s(d)} \cdot (\text{\# of flops per state}). \tag{9}$$

Since all states are equally likely, the expected number of surviving states in the state set $\mathcal{S}_d$ is given by, say

$$E_s(d) = \sum_{j=1}^{n_s(d)} P\left(\mathcal{S}_d^{[j]} \text{ survives}\right) = n_s(d) \cdot P\left(\mathcal{S}_d^{[1]} \text{ survives}\right)$$

$$= n_s(d) \cdot P\left(\mathcal{C}_d(\mathcal{S}_d^{[1]}) \leq r_d^2\right) \tag{10}$$

where $\mathcal{S}_d^{[1]}$ is the state which corresponds to $[s_d, s_{d-1}, \ldots, s_{d-l+1}] = [-(L-1/2), \ldots, -(L-1/2)]$, and where the number of states is given by $n_s(d) = L^{\min(l,d)-1}$ (so, when $d \geq l$ and the channel memory "fills," the number of states is a constant, $n_s(d) = L^{l-1}$). In what follows, we determine $E_s(d)$ for $d \geq l$.[1] We start by finding the probability that, for a given transmitted sequence, state $\mathcal{S}_d^{[1]}$ survives the pruning. Consider the following thought experiment. Assume that the sequence $\mathbf{s}_t$ of length $d$ was transmitted and that $\mathbf{x} = H\mathbf{s}_t + \mathbf{v}$ is observed. We wish to determine the probability that for an arbitrary sequence $\mathbf{s}_a$ of length $d$, it holds that

$$r^2 \geq \|\mathbf{x} - H_d \mathbf{s}_a\|^2 = \|\mathbf{v} + H_d(\mathbf{s}_t - \mathbf{s}_a)\|^2$$
$$= \|\mathbf{v} + (\Sigma_t - \Sigma_a)\mathbf{h}\|^2 \tag{11}$$

where the $d \times d$ matrix $H_d$ and the $l$-dimensional vector $\mathbf{h}$ are given by

$$\mathbf{H}_d = \begin{bmatrix} h_1 & & & & \\ \vdots & \ddots & & 0 & \\ h_l & \cdots & & h_1 & \\ & \ddots & & & \ddots \\ 0 & & h_l & \cdots & h_1 \end{bmatrix}, \quad \mathbf{h} = \begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_l \end{bmatrix}.$$

Furthermore, the structure of the sequence vectors $\mathbf{s}_t$ and $\mathbf{s}_a$ is of the form

$$\mathbf{s}_t = [\underbrace{s_{t,1} \cdots s_{t,d-l}}_{\mathbf{s}_{t,\text{past}}} | \underbrace{s_{t,d-l+1} \cdots s_{t,d}}_{\mathbf{s}_{t,\text{state}}}]^*$$

$$\mathbf{s}_a = [\underbrace{s_{a,1} \cdots s_{a,d-l}}_{\mathbf{s}_{a,\text{past}}} | \underbrace{s_{a,d-l+1} \cdots s_{a,d}}_{\mathbf{s}_{a,\text{state}} = -\frac{L-1}{2} \ldots - \frac{L-1}{2}}]^*$$

---

[1]For the other case $(d < l)$, $E_s(d)$ can be either found similarly, or simply approximated by the total number of states. This will be a good approximation, since this early in the trellis/tree there is not much pruning, and most of the states survive.

while the $d \times l$ dimensional matrices $\Sigma_t$ and $\Sigma_a$ are given by

$$\Sigma_t = \begin{bmatrix} s_{t,1} & & & 0 \\ \vdots & \ddots & & \\ s_{t,d-l+1} & \cdots & s_{t,d-2l} \\ \vdots & & & \vdots \\ s_{t,d} & \cdots & s_{t,d-l+1} \end{bmatrix}$$

$$\Sigma_a = \begin{bmatrix} s_{a,1} & & & 0 \\ \vdots & \ddots & & \\ s_{a,d-l+1} & \cdots & s_{a,d-2l} \\ & \ddots & & \vdots \\ \Sigma_{a,-\frac{L-1}{2}} & & s_{a,d-l+1} \end{bmatrix}$$

where the $\Sigma_{a,-(L-1)/2}$ region in $\Sigma_a$ has all entries equal to $-(L-1)/2$. The vector $[\mathbf{v}\,\mathbf{h}]^*$ in (11) is Gaussian, and hence, we can find the characteristic function of $\|\mathbf{v} + (\Sigma_t - \Sigma_a)\mathbf{h}\|^2$ (see [4])

$$\Phi(\omega) = \frac{\sqrt{2^{d+l}}}{\sqrt{(1 + j\omega 2\sigma^2)^{d-l} \prod_{i=1}^{l}[1 + j\omega 2(\sigma^2 + \rho_i)]}}$$

where $\rho_i, i = 1, \ldots, l$ are the eigenvalues of the matrix $(\Sigma_t - \Sigma_a)^*(\Sigma_t - \Sigma_a)$. Thus, for a given $(\mathbf{s}_t; \mathbf{s}_{a,\text{past}})$, the probability that the state $\mathcal{S}_d^{[1]}$ corresponding to $\mathbf{s}_{a,\text{state}} = [-(L-1)/2 \cdots -(L-1)/2]$ survives is given by

$$P\left(\mathcal{C}_d\left(\mathcal{S}_d^{[1]}\right) \le r_d^2 \Big| \mathbf{s}_t, \mathbf{s}_{a,\text{past}}\right)$$
$$= \frac{1}{2\pi} \int_{t=0}^{r^2} \int_{\omega=-\infty}^{\infty} \Phi(\omega) e^{-j\omega t} d\omega dt. \quad (12)$$

To find the probability $P(\mathcal{C}_d(\mathcal{S}_d^{[1]}) \le r_d^2)$, we need to average (12) over all $(\mathbf{s}_t; \mathbf{s}_{a,\text{past}})$

$$P\left(\mathcal{C}_d\left(\mathcal{S}_d^{[1]}\right) \le r_d^2\right)$$
$$= \frac{1}{L^{l-1}} \frac{1}{L^{2(d-l+1)}} \sum_{\mathbf{s}_{t,\text{state}}} \sum_{(\mathbf{s}_{t,\text{past}}; \mathbf{s}_{a,\text{past}})} \frac{1}{2\pi}$$
$$\times \int_{t=0}^{r^2} \int_{\omega=-\infty}^{\infty} \Phi(\omega) e^{-j\omega t} d\omega dt. \quad (13)$$

The outer summation in (13) is performed over $L^{l-1}$ states in which the transmitted sequence may terminate. The inner summation is performed over $L^{2(d-l+1)}$ possible pairs $(\mathbf{s}_{t,\text{past}}; \mathbf{s}_{a,\text{past}})$. An efficient enumeration of the symbol sequences that might ease the computation (13) by counting pairs $(\mathbf{s}_t; \mathbf{s}_{a,\text{past}})$ with the same $P(\mathcal{C}_d(\mathcal{S}_d^{[1]}) \le r_d^2|\mathbf{s}_t, \mathbf{s}_{a,\text{past}})$ and lead to a more explicit closed-form expression for $P(\mathcal{C}_d(\mathcal{S}_d^{[1]}) \le r_d^2)$, so far appears hard to obtain. Thus, we leave (13) in its current form to be used numerically in evaluating (9) and (10).

## IV. SIMULATION RESULTS AND SUMMARY

We consider a channel of length $l = 6$, transmitting 4-phase-shift keying (PSK)-modulated $(L = 4)$ data in blocks of length $T = 20$, and employ the combined VA/SD for ML detection at the receiver. Let us define the complexity exponent as $e =$
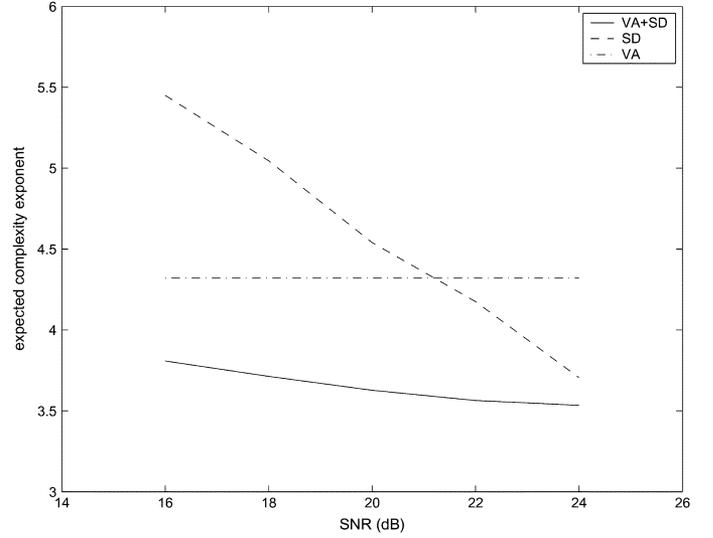


Fig. 3. Expected complexity exponents as a function of SNR for the VA, the SD, and the combined VA/SD, $l = 6$, $T = 20$, $L = 4$.
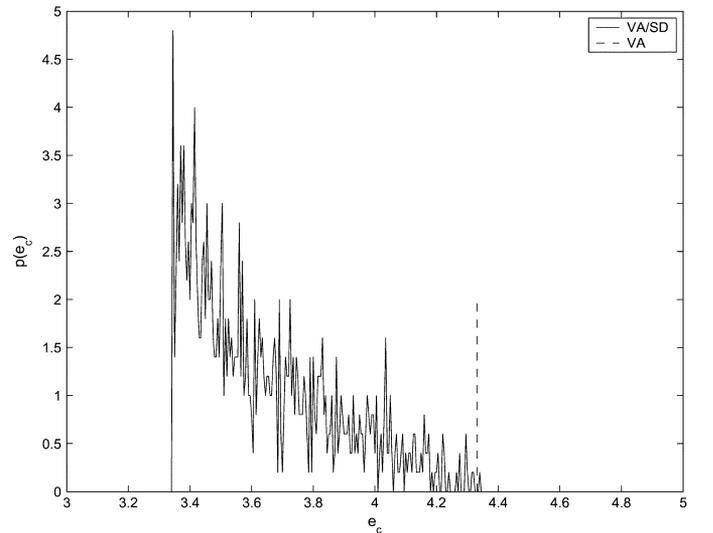


Fig. 4. Probability density function (empirical) of the complexity exponent of the combined VA/SD algorithm, $l = 6$, $T = 20$, $L = 4$, SNR $= 18$ dB. The dashed line denotes complexity exponent of the VA.

$\log_T F$, where $F$ denotes the total number of operations (flop count) performed when detecting $\mathbf{s}$. Fig. 3 shows the (empirically obtained) expected complexity exponent as a function of SNR. In the considered SNR range, the combined algorithm has lower expected complexity than both the SD and the VA. Furthermore, the expected complexity is roughly cubic in the considered range of SNR. If we further decrease SNR, the expected complexity of the combined algorithm increases; however, it is bounded above by the complexity of the VA.

Fig. 4 shows empirical distribution of the complexity exponent at SNR $= 18$ dB. The complexity is for the scheme that finds the optimal solution. In particular, we set the sphere radius so that a point is found with probability .9; if no point is found, we increase this probability to .99, and so on. As evident from Fig. 4, the complexity exponent of the VA/SD algorithm

is often significantly smaller than the complexity exponent corresponding to the VA (denoted by the vertical dashed line).

In summary, we proposed combining the sphere-constrained search of the SD and the dynamic programming principles of the VA for ML detection for channels with memory. The hybrid algorithm is either the SD modified to speed up the search for the closest point in the lattice, or the VA with the imposed sphere constraints resulting in state reduction on the trellis. We found the analytic expression for the expected complexity of the algorithm and illustrated it via simulations. The algorithm has expected complexity which is polynomial in the data block length over a wide range of SNRs.

## REFERENCES

[1] A. D. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Trans. Inf. Theory*, vol. IT-13, no. 1, pp. 260–269, Jan. 1967.

[2] 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.

[3] W. H. Mow, "Maximum-likelihood sequence estimation from the lattice viewpoint," *IEEE Trans. Inf. Theory*, vol. 40, no. 9, pp. 1591–1600, Sep. 1994.

[4] H. Vikalo and B. Hassibi, "On the sphere decoding algorithm. II. Generalizations, second-order statistics, and applications to communications," *IEEE Trans. Signal Process.*, vol. 53, no. 8, pp. 2819–2834, Aug. 2005.

[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] S. J. Simmons, "Breadth-first trellis decoding with adaptive effort," *IEEE Trans. Commun.*, vol. 38, no. 1, pp. 3–12, Jan. 1990.

[7] H. Zamiri-Jafarian and S. Pasupathy, "Complexity reduction of the MLSD/MLSDE receiver using the adaptive state allocation algorithm," *IEEE Trans. Wireless Commun.*, vol. 1, no. 1, pp. 101–111, Jan. 2002.