ACQUISITION PERFORMANCE OF BLIND SEQUENCE DETECTORS USING PER-SURVIVOR PROCESSING

Keith M. Chugg {chugg@milly.usc.edu}
Dept. of Electrical Engineering–Systems
University of Southern California
Los Angeles, CA 90089-2565

Abstract—The acquisition capabilities of binary sequence detection algorithms based on Per-Survivor Processing (PSP) is demonstrated via numerical results. The limits of these capabilities are motivated based on analytical results and verified via computer simulations. It is demonstrated that the acquisition of PSP-based data detection algorithms is very rapid and can be improved even further by carefully initialization.

I. INTRODUCTION

There are many applications where it is desirable to perform optimal or near-optimal mitigation/combining of intersymbol interference (ISI). Primary among these applications is digital mobile radio where ISI is caused by frequency-selective multipath fading. The random fading and the time-varying nature of this channel cause the gain associated with optimal data detection over a suboptimal equalization technique (e.g., Decision Feedback Equalization) to be significant [1]. This significant gain is in contrast to the telephone channel, for which DFE is highly effective, and may account for renewed interest in Maximum Likelihood Sequence Detection (MLSD) and Maximum A Posteriori (MAP) symbol detection algorithms.

We define “Conventional” Adaptive MLSD (CA-MLSD) as the combination of an MLSD processor based on the Viterbi Algorithm (VA), and a single channel estimator which is based on tentative decisions feedback from the MLSD processor after some tentative decision delay, which is typically less than the decoding depth used for the final decisions. The introduction of the concept of Per-Survivor Processing (PSP) [2], [3] was important because it provided a systematic approach to avoid the design trade-off associated with CA-MLSD. Namely, it is desirable to select the tentative delay large so that the tentative decisions are reliable, yet it is desirable to select the delay as small as possible so that the channel estimator is provided with up-to-date information. In a PSP-based adaptive MLSD algorithm, a channel estimate is maintained for each survivor sequence in the trellis and is updated conditioned on that data sequence. This intuitively-satisfying parallel decision feedback approach provides zero-delay decisions to the channel estimators, one of which agrees with the correct data. It has been shown that PSP may be viewed as a suboptimal approximation to an exhaustive data sequence tree-search, where the path metric used is computed based on per-path channel estimation [4], [5]. The notion of generalized-PSP was introduced in [6], [5]. A generalized-PSP algorithm is any algorithm which uses some recursive tree-search strategy (i.e., see [8]) using a metric based on per-path parameter estimation. There are many conceivable generalized-PSP algorithms, with the PSP-based “generalized VA” (i.e., multiple survivors per-state) investigated in [3], and the PSP-based M-Algorithm investigated in [7].

Most of the previous work in this area has concentrated on the performance of various adaptive data detection algorithms in the tracking mode. We define tracking as the mode of operation where the symbol error rate, averaged over the channel statistics, is time invariant. As discussed in [9], this is distinct from the acquisition mode, in which the receiver has little or no information about the ISI channel or the data sequence. In practice, the tracking mode is entered by periodically providing known training sequences. If the interval between training is too large with respect to the channel dynamics, the tracking assumption becomes invalid. In summary, there are two cases of practical interest: (i) the tracking mode with a time-varying channel, and (ii) the acquisition mode with a relatively static channel. Most likely due to the existing packet formats of GSM and IS-54, which contain adequate training signals, most of the research into quasi-optimal blind data detection algorithms has focused on the tracking mode. However, the capability of PSP and related recursive algorithms to acquire the channel blindly has been noted if not fully investigated. When acquisition has been studied, the performance has been characterized by the mean-square error of the channel estimate as a function of time (i.e., a learning curve). While this is useful, it does not translate in a simple manner to what is perhaps a more meaningful performance measure, namely the symbol error rate during acquisition and the probability of acquisition. These measures may be the most relevant when blind data detection algorithms are considered for application in next-generation packet mobile radio systems with little or no training overhead.

In this paper we characterize the acquisition performance of PSP and a class of related algorithms via numerical examples. In the process, the primary cause of misacquisition is motivated from analytical results and demonstrated through computer simulations.

This research has been funded in part by the Integrated Media Systems Center, a National Science Foundation Engineering Research Center with additional support from the Annenberg Center for Communication at the University of Southern California and the California Trade and Commerce Agency.
II. JOINT ML CHANNEL AND DATA ESTIMATION

The model under consideration is a discrete-time, symbol-spaced, signal defined by

$$z_i = \sum_{m=0}^{L-1} f_m a_{i-m} + w_i, \quad i = 0, 1, \ldots k$$  \hspace{1cm} (1)

where \( \{w_i\} \) is a white Gaussian sequence and \( f_m \) is an equivalent, unknown ISI channel. The digital data sequence \( \{a_i\} \) is assumed to the result of BPSK modulation so that \( a_i \in \{-1,+1\} \). As demonstrated in detail in [5], [9], the equivalent channel in the unknown case arises from front-end processing matched to the transmitted pulse and fractionally-spaced sampling is required to obtain a sufficient statistic for joint estimation of the channel and data. However, as will be seen, the performance limitations in the acquisition mode are not affected by sampling rate. Thus, in this paper we will consider the quantities in (1) to be scalars although the modification to fractional spacing is trivial based on the development in [5]. The vector version of the \( k+1 \) scalar equations in (1) is

$$\mathbf{z}_k = \mathbf{A}_k \mathbf{f}_k + \mathbf{w}_k,$$  \hspace{1cm} (2)

with

$$\mathbf{z}_k = \begin{bmatrix} z_k & z_{k-1} & \cdots & z_0 \end{bmatrix} \hspace{1cm} (3)$$

$$\mathbf{w}_k = \begin{bmatrix} w_k & w_{k-1} & \cdots & w_0 \end{bmatrix} \hspace{1cm} (4)$$

$$\mathbf{f} = \begin{bmatrix} f_{L-1} & f_{L-2} & \cdots & f_0 \end{bmatrix} \hspace{1cm} (5)$$

$$\mathbf{A}_k = \begin{bmatrix} a_{k-L+1} & a_{k-L+2} & \cdots & a_k \\ a_{k-L} & a_{k-L+1} & \cdots & a_{k-1} \\ \vdots & \vdots & \ddots & \vdots \\ a_{-L+1} & a_{-L+2} & \cdots & a_0 \end{bmatrix} \hspace{1cm} (6)$$

$$= \begin{bmatrix} \mathbf{\alpha}_k \end{bmatrix} = \begin{bmatrix} \mathbf{\alpha}_{k-1} \end{bmatrix} = \begin{bmatrix} \cdots \mathbf{\alpha}_0 \end{bmatrix}^T. \hspace{1cm} (7)$$

The metric to be minimized for joint maximum likelihood estimation of \( \{a_i\} \) and \( \mathbf{f} \) is the residual least-squares error of the per-sequence channel estimate \(^1\)

$$\Lambda_k(\tilde{\mathbf{A}}_k) = \| \bar{\mathbf{Q}}_k \mathbf{z}_k \|^2,$$  \hspace{1cm} (8)

where \( \bar{\mathbf{Q}}_k \) is the projection matrix onto the orthogonal complement of the range of \( \tilde{\mathbf{A}}_k \)

$$\bar{\mathbf{Q}}_k = \mathbf{I} - \tilde{\mathbf{A}}_k \tilde{\mathbf{A}}_k^T. \hspace{1cm} (9)$$

The pseudo-inverse of \( \tilde{\mathbf{A}}_k \) is denoted by \( \tilde{\mathbf{A}}_k^+ \) and is given by \( \tilde{\mathbf{A}}_k^+ = (\tilde{\mathbf{A}}_k^T \tilde{\mathbf{A}}_k)^{-1} \tilde{\mathbf{A}}_k^T \) when \( \tilde{\mathbf{A}}_k \) is rank \( L \). \(^2\) For issues of numerical stability and performance, we consider the generalization of (8) and (9) defined by a weighted least squares fit

$$\Lambda_k(\tilde{\mathbf{A}}_k) = \| \mathbf{W}_k^{1/2} \bar{\mathbf{Q}}_k \mathbf{z}_k \|^2$$  \hspace{1cm} (10)

\(^1\)The notation \( \tilde{m} \) is used to denote a hypothesized version of the quantity \( m \).

\(^2\) We assume that \( k \geq L - 1 \) so that the system is overdetermined.

where \( \mathbf{W}_k = \text{diag}(1, \rho, \rho^2, \ldots, \rho^k) \). The associate per-sequence channel estimate is \(^3\)

$$\hat{\mathbf{f}}_k = \hat{\mathbf{f}}_k(\tilde{\mathbf{A}}) = \tilde{\mathbf{A}}^T \mathbf{z}_k.$$  \hspace{1cm} (11)

The metric in (10) can be computed recursively via [5], [6]

$$\Lambda_k(\hat{\mathbf{A}}_k) = \rho \Lambda_{k-1}(\hat{\mathbf{A}}_{k-1}) + \frac{\rho}{\rho + \Delta_k} \| z_k - \tilde{\mathbf{A}}_k \hat{\mathbf{f}}_{k-1} \|^2$$  \hspace{1cm} (12)

with the associated per-sequence recursive least squares (RLS) channel estimate computed by

$$\hat{\Delta}_k = \tilde{\mathbf{a}}_k^T \hat{\mathbf{B}}_{k-1} \tilde{\mathbf{a}}_k$$ \hspace{1cm} (13a)

$$\hat{\mathbf{g}}_k = \hat{\mathbf{B}}_{k-1} \tilde{\mathbf{a}}_k \rho + \hat{\Delta}_k$$ \hspace{1cm} (13b)

$$\hat{\mathbf{f}}_k = \hat{\mathbf{f}}_{k-1} + \hat{\mathbf{g}}_k (z_k - \tilde{\mathbf{a}}_k^T \hat{\mathbf{f}}_{k-1})$$ \hspace{1cm} (13c)

$$\hat{\mathbf{B}}_k = (\hat{\mathbf{A}}_k^H \mathbf{W}_k \hat{\mathbf{A}}_k)^{-1} = \rho^{-1}(1 - \hat{\mathbf{g}}_k \hat{\mathbf{a}}_k^T) \hat{\mathbf{B}}_{k-1}.$$ \hspace{1cm} (13d)

Thus, a generalized-PSP algorithm for a static, unknown ISI channel is defined by the above recursion along with tree-search algorithm.

A. Equivalent Sequences

It is clear from the metric in (8) (i.e., \( \rho = 1 \)) that if two data matrices have the same range space (i.e., the same \( \mathbf{Q} \) matrix), that they will have identical metrics regardless of the observation \( \mathbf{z}_k \). We refer to such sequences as indistinguishable or equivalent sequences. The most simple example of two equivalent sequences is \( \hat{\mathbf{A}}_k \) and \( -\hat{\mathbf{A}}_k \), for which \( \mathbf{f}(\hat{\mathbf{A}}_k) = -\mathbf{f}(-\hat{\mathbf{A}}_k) \). For this simple case, differential encoding and decoding of the data may be used to eliminate the possible data inversion. However, more complex examples of equivalent sequences exist. Denote the class of all sequences that are equivalent to \( \hat{\mathbf{A}}_k \) by \( \mathcal{E}(\hat{\mathbf{A}}_k) \). Some asymptotic results on the size of these equivalence classes have been obtained in [6], [10]. The most relevant result is that, for large \( k \), the probability that \( \mathcal{E}(\hat{\mathbf{A}}_k) \) will consist of only \( \hat{\mathbf{A}}_k \) and \(-\hat{\mathbf{A}}_k \) goes to one. Intuitively this motivates the use of a tree-search strategy which prolongs the elimination of any sequence for as long as possible.

Since two equivalent hypothesized sequences \( \hat{\mathbf{A}}_k^{(1)} \) and \( \hat{\mathbf{A}}_k^{(2)} \) have the same range, the following hold

$$\hat{\mathbf{A}}_k^{(1)} = \hat{\mathbf{A}}_k^{(2)} \mathbf{M}_{1 \hat{2}}$$ \hspace{1cm} (14)

$$\hat{\mathbf{f}}_k^{(2)} = \mathbf{M}_{1 \hat{2}} \hat{\mathbf{f}}_k^{(1)}.$$ \hspace{1cm} (15)

\(^3\) The dependence of per-sequence channel estimates and the associated recursion quantities on the hypothesized sequence is suppressed for compactness.

\(^4\) This is for data matrices with rank \( L \), which is assumed since it can be shown that the number of data matrices with smaller rank is small and does not change with \( k \).
path associated with \( \mathbf{A}_k \) in favor of another member of \( \mathcal{E} \left( \mathbf{A}_k \right) \), thus eliminating the associated channel estimate as well.

The following two sequences are equivalent for \( L = 4 \)
\[
(\mathbf{a}_k = [ a_k \ a_{k-1} \ \cdots \ a_{1-L} ]^T)
\]
\[
\tilde{\mathbf{a}}^{(1)}_4 = [-1 \ -1 \ -1 \ -1 \ +1 \ +1 \ +1 ]^T
\]
\[
\tilde{\mathbf{a}}^{(2)}_4 = [-1 \ -1 \ -1 \ +1 \ +1 \ +1 \ -1 ]^T
\]
\[
(16)
\]
The decimal values of these sequences\(^5\) are (15)\(_{10}\) and (30)\(_{10}\), respectively. The associated channel estimates are related via
\[
\mathbf{M}_{1,2} = \begin{bmatrix}
0 & 0 & 0 & -1 \\
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0
\end{bmatrix}
\]
\[
(17)
\]
which implies that
\[
\hat{\mathbf{f}}^{(2)} = \mathbf{M}_{1,2} \hat{\mathbf{f}}^{(1)} = \begin{bmatrix}
-\hat{f}^{(1)}_0 \\
\hat{f}^{(1)}_0 \\
\hat{f}^{(1)}_1 \\
\hat{f}^{(1)}_1
\end{bmatrix}
\]
\[
(18)
\]
This particular equivalence class contains the sequences \{15, 30, 45, 60, 75, 90, 105, 120\}\(_{10}\) as well as all the compliments.

Through numerical experiments, it has been observed that these equivalence classes manifest themselves by causing a PSP-based algorithm to converge to shifted version of the actual transmitted sequence. For example, consider the case in (16) with the assumption that \( \mathbf{a}_4 = \tilde{\mathbf{a}}^{(1)}_4 \). If \( \tilde{\mathbf{a}}^{(2)}_4 \) is eliminated in favor of the equivalent \( \tilde{\mathbf{a}}^{(2)}_4 \), the PSP channel estimator will converge to
\[
\hat{\mathbf{f}} = \begin{bmatrix}
0 \\
\hat{f}_5 \\
\hat{f}_2 \\
\hat{f}_1
\end{bmatrix}
\]
\[
(19)
\]
with the associated data estimate being \( \hat{a}_k = a_{k-1} \). The effect of this convergence to a shifted version of the data is not necessarily catastrophic in many applications. For example, if there is external frame synchronization information available, or if the data sink is relatively insensitive to small shifts in the data sequence (e.g., an audio signal) then the effect is a loss in the signal-to-noise ratio and some additional noise from the channel estimation process.

III. Generalized-PSP: The “Multiple-State” Example

As briefly described in Section I, generalized PSP is simply the extension of per-path parallel decision feedback to an arbitrary tree-search algorithm. In order to improve acquisition performance, the number of sequences eliminated during the early stages of convergence should be minimized. An excellent summary of tree-search algorithms is contained in [8], where they are classified as depth-first (e.g., the Fano algorithm), breadth-first (e.g., the M-algorithm), and metric-first (e.g., the stack algorithm). It is difficult to quantify which approach is the best in terms of a complexity-performance trade-off. In fact, the ultimate drivers are most likely the implementation (i.e., hardware design methodology) and the desired characteristics of the data link (e.g., delay requirements). The purpose of introducing a tree-search algorithm other than the VA in this paper is to demonstrate how the bit error rate (BER) performance during acquisition can be improved by adding complexity to the tree-search.

The multiple state technique [10] simply increases the number of survivors maintained by constructing a trellis based on a memory length \( L_t \) in place of \( L \), with \( L_t > L \). For the binary modulation under consideration, the number of survivors is increased from \( 2^{L-1} \) in the VA to \( 2^{L_t-1} \). The term “multiple state” is used to convey the idea that for every channel state, defined by \( \mathbf{s}_k = [ a_{k-1} \ a_{k-2} \ \cdots \ a_{k-L_t+1} ] \), there are \( 2^{L_t-L} \) states in the expanded trellis. This concept is illustrated in Fig. 1, where a 4-state trellis is expanded to 8-states. The transition metric is still computed based on the actual memory parameter \( L \). This concept is very similar to the multiple survivor technique. However, the multiple state search is strictly trellis-based; meaning that, like the VA, it requires no sorting. In contrast, the multiple survivor approach, while also breadth-first, is not strictly trellis-based and, like the M-algorithm, requires some sorting. The multiple survivor approach is more flexible in the sense that the number of paths retained can be adjusted in increments of \( 2^{L-1} \), whereas the number of paths in the multiple state trellis is increased by powers of two. The multiple state approach may be viewed as multiple survivors with the constraint that the survivors maintained at each state have recent pasts which differ.

IV. Simulation Results

There are various assumptions that can be made regarding the initial knowledge of the receiver and states of the system. For example, it may be known which channel tap is the largest so that the channel estimates can be initialized to a Kronecker delta at this tap. Another distinction is what is assumed regarding the initial state of the channel. For example, if \( a_i = 0 \) for \( i < 0 \) is assumed, there is an implicit assumption that there is some frame synchronization information and that there is a temporal guard-band between data bursts. In all of the results presented in this paper, we assume that the channel is filled with random data and that nothing is known about the channel taps. Thus, while the modulation format and \( L \) are assumed to be known, the initial channel estimates are set to zero. In this sense the results may be pessimistic depending on the application (i.e., if additional information is available, its use will improve the performance).

\(^5\)We associate \( a_i = -1 \) with binary zero.
The numerical results presented are for an $L = 3$ channel with $\mathbf{f} = [0.408 \ 0.817 \ 0.408]^T$. The operating SNR is $||f||^2/\sigma_w^2 = 10$ dB. Simulations were run with other parameters with qualitatively similar results. The forgetting factor $\rho$ was taken to be 0.9. The trade-off observed with $\rho$ was that values too close to 1 caused more misacquisitions, while smaller values slowed convergence and resulted in a larger BER; $\rho = 0.9$ was found to provide a reasonable compromise. All simulations are based on differential encoding and decoding, with the known-channel performance shown as a lower bound on the BER. Each simulation was run by starting the algorithm and decoding the data from the first symbol. An estimate of the BER at different symbol positions was kept until at least 100 errors were observed at each position.

Fig. 2 shows the acquisition performance of a standard PSP algorithm (i.e., the 4-state trellis) based on the metric of (12). This performance is impressive when compared to that of conventional “blind equalization” algorithms with the known channel performance approached after 175 symbols. Fig. 3 illustrates the effects of misacquisition on the BER during acquisition. Specifically, the percentage of packets observed with $n$ errors is plotted versus $n$. Note that for a known channel this would be a Binomial distribution, but for the PSP algorithm there is a bimodal distribution suggesting that the misacquisitions (i.e., the packets with roughly half of the bits in error) dominate the performance.

These misacquisitions are in large part represented by the PSP algorithm locking onto a shifted version of the transmitted sequence, as predicted by the discussion in Section II-A. This is demonstrated by defining two new performance measures. First, the decoded data is compared against $\{a_k \pm m\}$ for each shifted sequence defined by $m = -L, \ldots, 0, \ldots, L$ and the version which best matches the estimated data is identified. If the zero shift (i.e., $m = 0$) is the best shift, the trial is declared to be a successful acquisition. If there is external frame synchronization or if the data sink is insensitive to shifts, the appropriate BER is the figure computed over the best shift $P_e$(synch). It is also interesting to note what the performance is for those trials when successful acquisition is declared, which we denote by $P_e$(acqd). In Fig. 4, these BER figures as well as the standard BER (i.e., the same as that in Fig. 2) are plotted – for this case, approximately 76% of the trials resulted in a success acquisition. Given that acquisition occurs, the known channel performance is approached after approximately 30 bits.

The multiple state algorithm PSP algorithm was also simulated under the same conditions. The performance of an $L_t = 4$ multiple state PSP algorithm is plotted in Fig. 5. The performance of this 8-state algorithm is no better than that of the 4-state standard PSP algorithm. This is due to a phenomenon that we call trellis splitting.

An even number of errors is more likely due to the differential encoding/decoding.
Trellis splitting occurs when for each state $s$, the complement state $\bar{s}$ has a channel estimate which differs only by sign. In this case, the PSP-based algorithm will effectively split the trellis by maintaining two sets of survivors, differing only by sign, which do not interact. When the 8-state algorithm is initialized with zero metric for each state, as in the simulations of Figures 2–4, the PSP algorithm falls directly into this partitioned trellis condition.\footnote{This actually has to do with the way that “ties” are handled in the add-compare-select process as well.} This occurs because the first few steps through the trellis do not depend on the value of the received signal when the initial channel estimates are set to zero. This effect also requires a consistency check when a finite decoding depth is used (i.e., one must verify that the survivor being traced back on is consistent with previous decisions).

Since differential encoding is used, it may be assumed that $a_{-2} = -1$ without loss of generality. Thus, only two states need be initialized with a zero metric and the other states can be initialized with a large value and thus eliminated. This initialization technique reduces the likelihood that the trellis will split. This is demonstrated in Fig. 5, where the curve labeled “smart initialization” shows an improvement over the $L_t = L$ standard PSP performance. The performance improvement is significantly greater for $L_t = 5$ and $L_t = 6$ as is illustrated in Fig. 6.

**V. Conclusion**

We have demonstrated that purely-recursive, PSP-based algorithms can converge to approximately the known-channel MLSD performance within as few as fifteen symbols. These results were obtained without assuming any knowledge of the signal except for the length of the ISI and the modulation format. It is possible that additional knowledge, which may be available in a cooperative environment, will further enhance these capabilities. Thus, such algorithms may be valuable for reducing the considerable training overhead in mobile radio packet formats in future systems.

\footnote{This actually has to do with the way that “ties” are handled in the add-compare-select process as well.}