Sequence Detection and Equalization for Pulse-Position Modulation

John R. Barry

School of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, Georgia, 30332-0250

ABSTRACT

We propose several strategies for detecting digital pulse-position modulation (PPM) in the presence of intersymbol interference and additive white Gaussian noise. We develop an equivalent vector channel by viewing PPM as a binary block code, and use this vector channel to derive the ML sequence detector. We also propose three sub-optimal block-by-block detectors, all of which use block decision-feedback equalization (DFE) to mitigate inter-block interference, but differ in how they mitigate intra-block interference: the ML-BDFE accounts for it in an optimal way, the comparator-BDFE uses linear equalization, and the correcting-scalar-DFE uses DFE. We illustrate low-complexity implementations of these equalizers, and compare their performance to the ML detector using Monte-Carlo simulations.

I. INTRODUCTION

Digital pulse-position modulation (PPM) is widely used in intensity-modulation optical communication systems, such as fiber-optic [1] and satellite systems [2], primarily because of its high average-power efficiency [3][4]. The abundance of bandwidth in these applications makes the poor bandwidth efficiency of PPM of little concern. In contrast, the bandwidth of indoor wireless infrared channels is severely constrained due to multipath optical propagation [5][6]. The objective of this work is to develop detection strategies for mitigating intersymbol-interference (ISI) so as to extend the usefulness of PPM to channels with severe ISI.

PPM can be viewed as a simple nonlinear block code; specifically, the rate $\log_2 L/L$ block code consisting of the set of L binary L-tuples with unity Hamming weight. Therefore, equalizers designed for general block codes, such as the recent work reported in [7]—[10], can be applied to PPM. On the other hand, because the only feature of PPM exploited in this paper is that PPM is a block code, our results apply directly to all binary and non-binary block codes, including multidimensional-constellation codes.

In the next section we describe an equivalent vector (or multiple-input/multiple-output; MIMO) channel model for PPM on ISI channels; it results by first viewing the PPM signal as a cyclostationary binary PAM signal, employing a whitened-matched filter front end, and then grouping the coded bits into blocks of length L. Assuming the channel has finite memory, the vector channel output becomes a noisy observation of a finite-state machine driven by a white symbol sequence, allowing application of the Viterbi algorithm to determine the maximum likelihood (ML) solution, as described in Sect. III.

In Sect. IV we propose sub-optimal equalization structures that are less complex than the ML sequence detector. Building on the theory of MIMO equalization developed elsewhere [7]—[14], we first examine the block DFE (BDFE) [8]—[11], which operate on the vector channel, thus requiring matrix multiplications. We then show how the BDFE can be implemented via a simple modification of a scalar equalizer. We also propose other scalar equalization structures, and compare their performance with that of the ML sequence detector.

II. EQUIVALENT VECTOR CHANNEL

A model for L-PPM transmission over channels with ISI is shown in Fig. 1. At the transmitter, a parallel-to-serial convertor converts a sequence of PPM vector blocks $\{\boldsymbol{x}_k\}$ with block rate $1/T$ into a single scalar "chip" sequence $\{x_j\}$ with chip rate L/T, where $\boldsymbol{x}_k = (x_{kL}, \ldots, x_{kL+L-1})'$. For each k, only one of the components of $\boldsymbol{x}_k$ is unity, and the other $L-1$ are zero. The chip sequence drives a transmit filter with pulse shape $p(t)$, so that the transmitted signal is given by:

$$x(t) = \sum_{j=-\infty}^{\infty} x_j\, p(t - jT/L)\,. \tag{1}$$

The received signal $y(t)$ is then given by:

$$y(t) = \sum_{j=-\infty}^{\infty} x_j\, g(t - jT/L) + n(t)\,, \tag{2}$$

where $g(t)$ is the convolution of $p(t)$ with the channel impulse response $b(t)$, and $n(t)$ is additive white Gaussian noise with two-sided power spectral density (PSD) N_0. Signals of the form (2) may be filtered by Forney's whitened-matched filter and then sampled at the chip rate L/T without compromising receiver performance [15]. The resulting equivalent discrete-time channel is described by:

$$y_j = \sum_{m=-\infty}^{\infty} h_m\, x_{j-m} + n_j\,, \tag{3}$$

where the discrete-time impulse response h_j is causal, monic ($h_0 = 1$), and minimum phase, and where the noise sequence n_j is white and Gaussian with PSD $\sigma^2 \equiv N_0/\gamma$, where γ is the geometric mean of the folded spectrum:

$$\gamma = \exp\left\{\frac{T}{L}\int_{-L/2T}^{L/2T} \log\left(\frac{L}{T}\sum_{k=-\infty}^{\infty} |G(f - kL/T)|^2\right) df\right\}\,. \tag{4}$$

Supported in part by an IBM Faculty Development Award and by NSF under grant number NCR-9308968.

As appeared in IEEE International Conference on Communications, ICC´94, New Orleans, LA, vol. 3, pp. 1561-1565, May 1994.

As shown in Fig. 1-a, the samples y_j are grouped in blocks of length L using a serial-to-parallel convertor, yielding a vector-valued output $\boldsymbol{y}_k$, where $\boldsymbol{y}_k = (y_{kL}, y_{kL+1}, \ldots, y_{kL+L-1})'$.
The equivalent vector channel between $\boldsymbol{x}_k$ and $\boldsymbol{y}_k$, as shown in Fig. 1-b is, described by:

$$\boldsymbol{y}_k = \sum_{m=0}^{\infty} \boldsymbol{H}_m \boldsymbol{x}_{k-m} + \boldsymbol{n}_k , \tag{5}$$

where the channel impulse response is a Toeplitz sequence $\boldsymbol{H}_k$, with $[\boldsymbol{H}_k]_{ij} = h_{kL+i-j}$:

$$\ldots \boldsymbol{H}_{-2} = \boldsymbol{H}_{-1} = \boldsymbol{0}, \quad \boldsymbol{H}_0 = \begin{bmatrix} & & & & 0 \\ & & & & \\ & & & 1 & \\ & & & h_1 & \\ & & & h_2 & \\ & & & \vdots & \\ & & & h_{L-1} & \end{bmatrix}, \ \textit{etc.}, \tag{6}$$

and the noise sequence $\boldsymbol{n}_k$ has autocorrelation function $\boldsymbol{R}_{\boldsymbol{n}}(m) = \mathrm{E}[\boldsymbol{n}_{k+m}\boldsymbol{n}_k'] = \sigma^2\delta_m\boldsymbol{I}$, resulting in a white spectrum: $\boldsymbol{S}_{\boldsymbol{n}}(z) = \sum \boldsymbol{R}_{\boldsymbol{n}}(m)z^{-m} = \sigma^2\boldsymbol{I}$. When the symbols $\boldsymbol{x}_k$ are chosen independently and uniformly from the PPM alphabet, the symbol spectrum is also white: $\boldsymbol{S}_{\boldsymbol{x}}(z) = (1/L)\boldsymbol{I}$.

III. ML SEQUENCE DETECTION

A computationally efficient ML sequence detector for MIMO channels of the form (5) was first derived in [16]. In our application, the ML sequence detector chooses the sequence of scalar chips $\{\hat{x}_j\}$, or equivalently the sequence of symbols $\{\hat{\boldsymbol{x}}_k\}$, that minimizes the following expressions [15][16]:

$$\sum_{j=0}^{\infty} | y_j - \sum_{m=0}^{\infty} h_m \hat{x}_{j-m} |^2 = \sum_{k=0}^{\infty} \| \boldsymbol{y}_k - \sum_{m=0}^{\infty} \boldsymbol{H}_m \hat{\boldsymbol{x}}_{k-m} \|^2 , \tag{7}$$

where $\| \boldsymbol{y}_k \|^2 = \Sigma_{j=0}^{L-1} | y_{kL+j} |^2$ denotes the squared Euclidean norm. Because x_j is the output of a block coder, not all sequences $\{x_j\}$ are allowable, so only allowable chip sequences $\{x_j\}$ are searched in the left-hand side of (7). When the input symbols $\boldsymbol{x}_k$ are chosen uniformly and independently from the PPM alphabet, on the other hand, as assumed here, every symbol sequence $\{\boldsymbol{x}_k\}$ is equally likely, so that all symbol sequences $\{\boldsymbol{x}_k\}$ are searched in the right-hand side. This property simplifies implementation, so we focus on the right-hand side of (7).

When the channel has finite memory μ, so that $\boldsymbol{H}_k = \boldsymbol{0}$ for $k > \mu$, the received sequence $\boldsymbol{y}_k$ of (5) represents a noisy observation of a finite-state machine driven by a white sequence $\boldsymbol{x}_k$. The ML sequence can then be found in the usual way [15]: by finding the minimum-length path through a trellis, which is best accomplished using the Viterbi algorithm [17]. The branch metric for a transition from state i to state j at time k is $\|\boldsymbol{y}_k - \boldsymbol{s}^{(i\to j)}\|^2$, where $\boldsymbol{s}^{(i\to j)}$ is the unique output associated with the transition from i to j. The probability of a symbol (block) error for the ML sequence detector is well approximated at high SNR by [15]:

$$P_e \approx Q(d_{min}/(2\sigma)), \tag{8}$$

where $\sigma^2 = N_0/\gamma$ is the variance of each component of $\boldsymbol{n}_k$, and d_{min} is the minimum distance between received sequences:

$$d_{min}^2 = \min_{\{\boldsymbol{e}_k\}} \sum_{k=0}^{\infty} \| \sum_{m=0}^{\infty} \boldsymbol{H}_m \boldsymbol{e}_{k-m} \|^2. \tag{9}$$

The above minimization is performed over all possible error sequences $\{\boldsymbol{e}_k\}$ starting at time zero, using an error alphabet of $\{\boldsymbol{u} - \boldsymbol{v} : \boldsymbol{u}, \boldsymbol{v} \in C\}$, where C is the set of valid PPM codewords.

The number of states in the trellis will be L^{μ}, where μ is the memory in the vector channel and is related to the memory M in the scalar channel by $\mu = \lceil M/L \rceil$. Note that, in addition to the add-compare-select computation of the scalar MLSE algorithm, the vector ML sequence detector requires an additional L squaring operations per branch computation. For large values of L and μ, therefore, the ML sequence detector may be computationally infeasible.

IV. EQUALIZATION

Equalization is a suboptimal strategy for detecting signals in the presence of ISI, its primary advantage over ML sequence detection being a reduction in complexity. We shall consider several block-rate (vector) equalizers and chip-rate (scalar) equalizers. In every case (see Fig. 2, Fig. 3, and Fig. 4), the equalizer is designed so that its output $\boldsymbol{w}_k$ at time k is in some sense close to the k-th transmitted symbol $\boldsymbol{x}_k$, and the receiver decides on $\hat{\boldsymbol{x}}_k$ using a memoryless decision device (slicer). A naive slicer chooses the $\hat{\boldsymbol{x}}_k$ closest in Euclidean distance to $\boldsymbol{w}_k$, but this is not generally optimal unless the components of the slicer error vector $\varepsilon_k = \boldsymbol{w}_k - \boldsymbol{x}_k$ are i.i.d. zero-mean Gaussian random variables. Nevertheless, the naive slicer is often preferred due to its simple implementation (the naive slicer reduces to an L-way comparator, or a "choose maximum" device, when the block code is PPM). In some cases, such as the BDFE of the next section, the optimal slicer can be found explicitly. In all cases, however, it is worth emphasizing that, unless the components of the error vector are i.i.d. Gaussian random variables, the mean-squared error (MSE = $\mathrm{E}\,\|\boldsymbol{w}_k - \boldsymbol{x}_k\|^2$) does not uniquely characterize error performance.

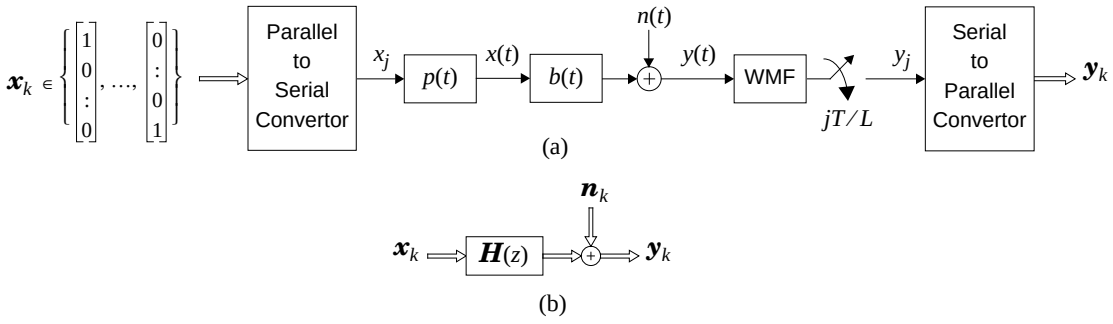

Fig. 1. Channel models for L-position PPM: (a) composite system; (b) equivalent vector (MIMO) channel.

IV-A. Block Equalizers

Since the channel of Fig. 1-b is a linearly filtered vector (MIMO) communication channel with AWGN, we may apply principles of MIMO equalization [7]—[14], referred to here as block equalization. The block equalization structure is shown in Fig. 2-a. It has the same form as a conventional scalar DFE. The received signal is passed through a forward filter with transfer function $\boldsymbol{C}(z) = \sum_k \boldsymbol{C}_k z^{-k}$. Decisions $\hat{\boldsymbol{x}}_k$ are fed through a feedback filter $\boldsymbol{D}(z)$, and the result is subtracted from the input of a vector slicer. The block linear equalizer results when $\boldsymbol{D}(z) = \boldsymbol{0}$.

The optimal zero-forcing BDFE filters may be derived as in the scalar case [7][18]. Specifically, if we limit consideration to filters that completely eliminate ISI, so that $\boldsymbol{w}_k$ equals $\boldsymbol{x}_k$ when the noise is zero, and then choose $\boldsymbol{C}(z)$ and $\boldsymbol{D}(z)$ to minimize the MSE $= \mathrm{E}\| \boldsymbol{w}_k - \boldsymbol{x}_k \|^2$ under the assumption that $\hat{\boldsymbol{x}}_k = \boldsymbol{x}_k$, the results are:

$$\boldsymbol{C}(z) = \boldsymbol{H}_0^{-1} \tag{10}$$

$$\boldsymbol{D}(z) = \boldsymbol{H}_0^{-1}\boldsymbol{H}(z) - \boldsymbol{I} \tag{11}$$

$$MSE_{BDFE} = \sigma^2 \mathrm{trace}(\boldsymbol{H}_0\boldsymbol{H}_0')^{-1}, \tag{12}$$

where a prime denotes conjugate transpose. Assuming correct decisions, $\hat{\boldsymbol{x}}_k = \boldsymbol{x}_k$, the slicer input is then $\boldsymbol{w}_k = \boldsymbol{x}_k + \boldsymbol{H}_0^{-1}\boldsymbol{n}_k$. In this case, it is easy to show that the optimal (maximum-likelihood) slicer chooses $\hat{\boldsymbol{x}}_k$ so as to minimize $\|\boldsymbol{H}_0(\boldsymbol{w}_k - \hat{\boldsymbol{x}}_k)\|^2$. If we define $\boldsymbol{H}_0(\boldsymbol{w}_k - \boldsymbol{x}_k)$ as the error for the ML slicer, then its mean-squared value is $MSE_{ML\text{-}BDFE} = L\sigma^2 \le MSE_{BDFE}$.

The optimal minimum-MSE BDFE can be derived in a similar way [7][12][14], and it would outperform the zero-forcing BDFE at low SNR. We shall not pursue it here, however, because our focus is on equalizers that are easily implemented.

IV-B. Scalar Equalizers

The block equalizers of the previous section, although reduced in complexity as compared to the MLSE, still appear quite complex. This motivates the investigation of chip-by-chip, scalar, or "random-phase" [7] equalizers.

The scalar zero-forcing linear equalizer (LE) is the most straight forward. The WMF samples y_j are passed through a scalar filter with transfer function $C(z) = 1/H(z)$, before the serial to parallel conversion, as illustrated in Fig. 3-a. Note that this scalar LE is equivalent to a block LE, which results when $\boldsymbol{C}(z) = \boldsymbol{H}(z)^{-1}$ and $\boldsymbol{D}(z) = \boldsymbol{0}$ in Fig. 2. The LE does not exist when $H(z)$ has poles on the unit circle.

A scalar zero-forcing DFE (SDFE) feeds back tentative "chip decisions" to cancel intra-block as well as inter-block interference, as illustrated in Fig. 3-b. Assuming that these chip decisions are correct, the MSE is given by:

$$MSE_{SDFE} = L\sigma^2. \tag{13}$$

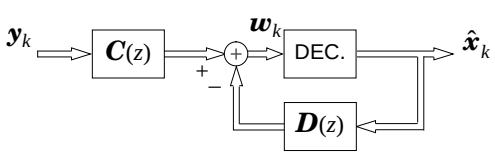

Fig. 2. The Block Decision-Feedback Equalizer (BDFE).

Comparing with (12), we might conclude that the SDFE outperforms the BDFE with naive slicer, equaling the performance of the BDFE with ML slicer. However, the chip-decisions are highly unreliable, and the resulting error propagation can reduce performance significantly. The true MSE of the SDFE will always exceed $L\sigma^2$, as illustrated by example in Sect. V.

IV-C. Correcting Scalar Equalizers

We next modify the scalar equalizers in a natural way to improve performance, yielding the so-called "correcting" scalar equalizers. The correcting scalar-LE turns out to be precisely equivalent to the vector BDFE, thus providing valuable insight into the BDFE's function, as well as suggesting a simple implementation.

The Correcting SLE: An Alternative View of the BDFE

The scalar zero-forcing linear equalizer of Fig. 3-a has transfer function $C(z) = 1/H(z)$. Since $H(z)$ is monic and causal, $C(z)$ can be implemented using linear feedback, as illustrated in Fig. 4 (ignoring the outer feedback loop for the moment), where the strictly causal feedback filter has transfer function $H(z) - 1$.

The contents of the memory elements in the delay line at time j are $\{w_{j-1}, w_{j-2}, \ldots, w_{j-M}\}$, where M is the length of the scalar channel impulse-response tail. It is well known that, since w_j represents a noisy estimate of the actual chip sequence x_j, the performance would improve if we fed back the true chip

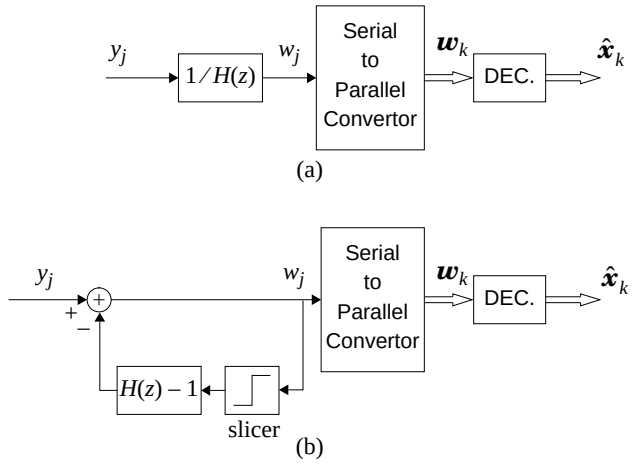

Fig. 3. Scalar Equalizers: (a) SLE, and (b) SCDFE.

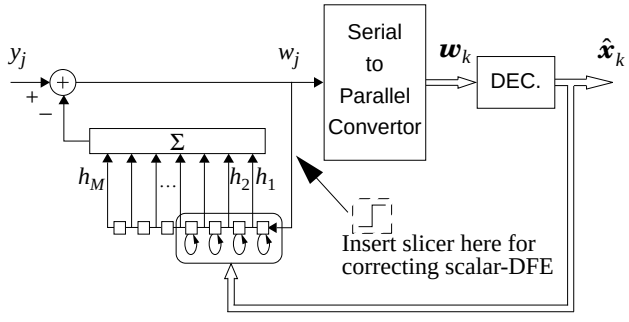

Fig. 4. The correcting scalar-LE, which implements the *block* DFE. Inserting a slicer as shown results in the correcting scalar-DFE.

sequence x_j rather than its noisy estimate w_j. (In other words, ideal decision-feedback equalizers outperform linear equalizers.) Unfortunately, there is a decoding delay (inherent in the serial-to-parallel convertor) of anywhere from one to L chips, preventing immediate decision feedback. Specifically, the decoding delay for the l-th chip in a block is $L - l$, where $l \in \{0, \ldots, L-1\}$.

We can still achieve some of the decision-feedback advantage, however, by feeding back the noisy estimates w_j when decisions are unavailable, and then "correcting" the delay line once a decision has been made. That is, when j is a multiple of L, the decisions about the previous L chips are known, and so the values stored in the first L delays may then be "corrected." In Fig. 4, the outer feedback path (between the decision device and the delay line) and the L circular arrows (assuming $L = 4$) are meant to represent this correction operation: a fraction of a chip[1] before time j is a multiple of L, the first L memory elements of the delay line $\{w_{j-1}, w_{j-2}, \ldots, w_{j-L}\}$ are replaced by $\{\hat{x}_{j-1}, \hat{x}_{j-2}, \ldots, \hat{x}_{j-L}\}$.

A closer look at the above-described correcting scalar-LE reveals that it implements precisely the vector BDFE of Sect. IV-A. The proof is omitted due to lack of space.

The Correcting Scalar-DFE

As in the LE case, it is also beneficial to correct the delay line for the scalar DFE. This leads to the correcting SDFE, as shown in Fig. 4 (with the slicer inserted as indicated). Intuitively, the only difference between the correcting SDFE and the BDFE is the way it mitigates intra-block interference; both mitigate inter-block interference using block decisions, but the correcting SDFE uses tentative chip-by-chip decisions to mitigate intra-block interference, whereas the BDFE uses linear equalization.

Although the correcting equalizers always outperform their non-correcting counterparts, comparisons between the BDFE (i.e., the correcting scalar-LE) and correcting scalar-DFE must be made on a case-by-case basis, depending on the channel and the slicer. This is illustrated further in the next section.

V. NUMERICAL RESULTS

As an example, consider the transmission of $L = 2$ PPM over a scalar channel with transfer function $H(z) = 1 - z^{-1} + z^{-2}$. The ISI in this channel is particularly severe, and cannot be overcome by an increase in signal power; even at infinite SNR, the error rate is $1/4$. From (9) we find that $d_{min} = \sqrt{10}$, and (8) approximates the error performance of the ML sequence detector. The dashed curve in Fig. 5 plots this estimate versus $\mathrm{SNR} \equiv 1/\sigma^2$. Simulation results for the ML sequence detector are represented in the figure by open circles, and were based upon 10^6 blocks (2×10^6 chips).

Fig. 5 also illustrates the performance of several equalization structures. (Since $H(z)$ has zeros on the unit circle, the zero-forcing LE does not exist.) The best performance is achieved by the BDFE with an ML slicer; it is about 3 dB from the ML sequence detector, and outperforms the BDFE with a comparator slicer by about 1 dB. These numbers agree with theory; neglecting error propagation, the slicer error at time k is:

$$\boldsymbol{H}_0^{-1}\boldsymbol{n}_k = [n_{kL}\ ,\ (n_{kL} + n_{kL+1})]^T. \tag{14}$$

The probability of error is then easily shown to be $Q(\sqrt{5}/2\sigma)$ and $Q(1/\sigma)$ for the BDFE with ML slicer and comparator slicer, respectively, so that the expected penalties are 3 and 4 dB, respectively.

Also included in Fig. 5 are simulation results for the scalar-DFEs, with and without correction. A significant gain of about 2.2 dB is achieved by the correcting scalar-DFE over its non-correcting counterpart. This implies that error propagation from the chip-by-chip slicer is significant. Although the BDFE with comparator slicer outperforms the correcting-DFE here, this is not always the case, as illustrated in the next example.

We now consider the same channel, but with $L = 8$ rather than $L = 2$. Simulation results are presented in Fig. 6. The dashed curve approximates the performance of the ML sequence detector from (8), using $d_{min} = 2$ as calculated from (9). (The approximation is poor at low SNR because there are 5 five pairs of sequences separated by a distance of d_{min}.) The other curves are the result of simulating 10^6 blocks. Unlike the previous example, here L is relatively large compared to the channel memory. Therefore, most of the ISI is confined to a single block, and the effects of inter-block interference are diminished. Since the BDFE with ML slicer mitigates intra-block interference in an optimal way, its performance is very close to that of the optimal ML sequence detector. On the other hand, the BDFE with comparator slicer essentially ignores the effects of intra-block interference, thus explaining its poor performance. The correcting SDFE is only negligibly better than its non-correcting counterpart; this is because the variable decoding delay usually exceeds the channel memory, so that most of the corrections have no effect.

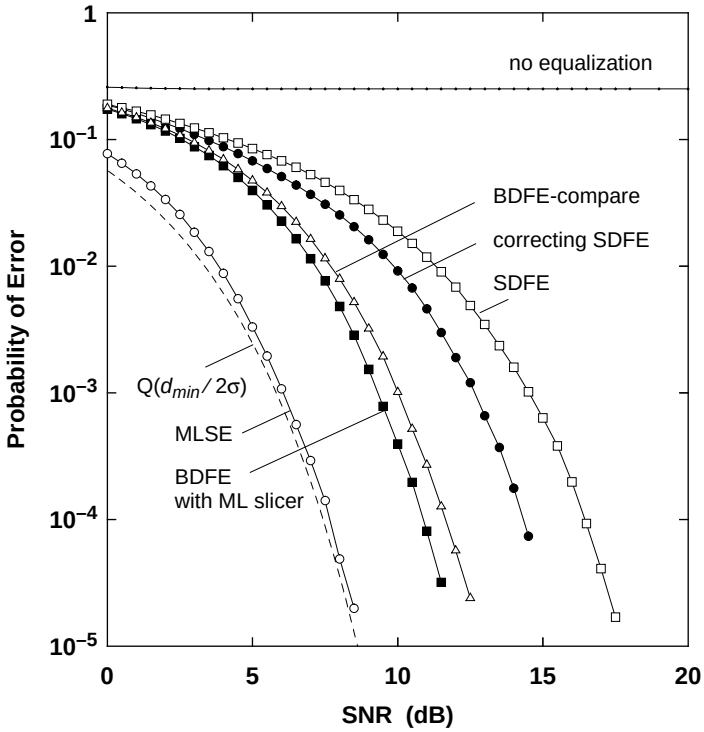

Fig. 5. Performance for $L = 2$ PPM with $H(z) = 1 - z^{-1} + z^{-2}$.

1. The corrections are made asynchronously, after chip number $kL - 1$ and before chip number kL, for every integer k.

VI. CONCLUSIONS

With the aid of an equivalent vector channel model, the maximum likelihood sequence detector for any block code subject to intersymbol interference and additive white Gaussian noise may be implemented using the Viterbi algorithm in a straight forward manner. The complexity of the ML sequence detector is high, however, and so reduced-complexity detectors are desirable. We examined the performance of several block-by-block equalization schemes, some of which required no matrix multiplications. These equalizers were effective at mitigating ISI, although for some channels there remained a significant gap between their performance and that of the ML sequence detector. Future research should examine the effectiveness of reduced-complexity detectors at closing this gap.

VII. REFERENCES

[1] I. Garrett, M. Calvert, M. Sibley, R. Unwin, and R. Cryan, "Optical Fibre Digital Pulse-Position Modulation," *Br. Telecom. Technol. Journal*, vol. 7, no. 3, July 1989.

[2] T. Wiesmann, S. Borner, and G. Ohm, "Optical QPPM Communication Subsystem for Free-Space Links," *Space Communications*, vol. 7, pp. 443-450, November 1990.

[3] J. R. Pierce, "Optical Channels: Practical Limits with Photon Counting," *IEEE Transactions on Communications*, vol. 26, no. 12, pp. 1819-1821, December 1978.

[4] R. J. McEliece, "Practical Codes for Photon Communication," *IEEE Transactions on Information Theory*, vol. IT-27, no. 4, pp. 393-398, July 1981.

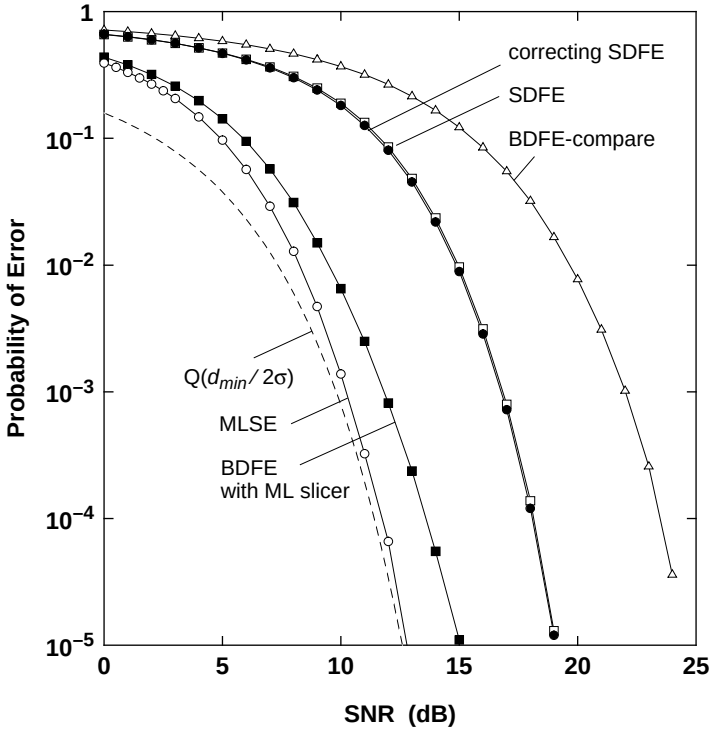

Fig. 6. Performance for L = 8 PPM with $H(z) = 1 - z^{-1} + z^{-2}$.

[5] J. M. Kahn, W. J. Krause, and J. B. Carruthers, "Experimental Characterization of Non-Directed Indoor Infrared Channels," submitted to *IEEE Transactions on Communications*, 1993.

[6] J. R. Barry, J. M. Kahn, W. J. Krause, E. A. Lee, D. G. Messerschmitt, "Simulation of Multipath Impulse Response for Indoor Wireless Optical Channels," *IEEE Journal of Selected Areas in Communications*, vol. 11, no. 3, 367-379, April 1993.

[7] A. Duel-Hallen, "Equalizers for Multiple Input/Multiple Output Channels and PAM Systems with Cyclostationary Input Sequences," *IEEE Journal on Selected Areas in Communications*, vol. 10, no. 3, pp. 630-639, April 1992.

[8] S. Kasturia and J. M. Cioffi, "Vector Coding with Decision-Feedback Equalization for Partial Response Channels," *IEEE GLOBECOM*, paper 27.3, Hollywood, FL, November 1988.

[9] W. L. Abbott and J. M. Cioffi, "Combined Equalization and Coding for High-Density Saturation Recording Channels," *IEEE Journal on Selected Areas in Communications*, vol. 10, no. 1, pp. 168-181, January 1992.

[10] J. W. Lechleider, "The Optimum Combination of Block Codes and Receivers for Arbitrary Channels," *IEEE Transactions on Communications*, vol. 38, no. 5, pp. 615-621, May 1990.

[11] D. Williamson, R. A. Kennedy, and G. W. Pulford, "Block Decision Feedback Equalization," *IEEE Transactions on Communications*, pp. 255-264, vol. 40, no. 2, February 1992.

[12] M. L. Honig, P. Crespo, and K. Steiglitz, "Suppression of Near- and Far-End Crosstalk by Linear Pre- and Post-Filtering," *IEEE Journal on Selected Areas in Communications*, vol. 10, no. 3, pp. 614-629, April 1992.

[13] J. Salz, "Digital Transmission Over Cross-Coupled Linear Channels," *AT&T Technical Journal*, vol. 64, no. 6, pp. 1147-1159, July-August 1985.

[14] M. Kavehrad and J. Salz, "Cross-Polarization Cancellation and Equalization in Digital Transmission over Dually Polarized Multipath Fading Channels," *AT&T Technical Journal*, vol. 64, no. 10, pp. 2211-2245, December 1985.

[15] G. D. Forney, Jr., "Maximum-Likelihood Sequence Estimation of Digital Sequences in the Presence of Intersymbol Interference," *IEEE Transactions on Information Theory*, vol. IT-18, no. 3, pp. 363-378, May 1972.

[16] W. Van Etten, "Maximum Likelihood Receiver for Multiple Channel Transmission Systems," *IEEE Transactions on Communications*, pp. 276-283, February 1976.

[17] G. D. Forney, Jr., "The Viterbi Algorithm," *Proceedings of the IEEE*, vol. 61, no. 3, pp. 268-278, March 1973.

[18] E. A. Lee and D. G. Messerschmitt, *Digital Communication*, Second Edition, Kluwer, 1994.