Abstract—In this paper, we investigate the turbo equalization using an algorithm called probabilistic data association (PDA). We first propose a general structure for the PDA, which consists of a linear interference cancellation step followed by a probabilistic data association step in every iteration. Based on the general structure, we show that the original PDA belongs to one variation and it is computational inefficient. We then unveil that the popular soft linear MMSE (SLMMSE) equalizer can be considered as one sweep within a generalized PDA. Such connection implies that further performance improvement over the SLMMSE equalizer is possible if the PDA is applied instead in turbo equalization. We also provide a way for the PDA equalizer to incorporate the \textit{a priori} probability, which makes the PDA readily applicable to turbo equalization.

I. INTRODUCTION

Data transmission in many practical communication systems experiences intersymbol interference (ISI). ISI is one of the major obstacles to high data rate communications. To combat ISI, equalization is performed at the receiver. Due to the high computational complexity of the optimum equalizer, much effort has been devoted to developing inexpensive and effective suboptimal solutions that employ, for instance, zero forcing, linear minimum mean squared error (LMMSE), and decision feedback approaches [1]. In addition to equalization, ISI can be further suppressed through the use of an error control code (ECC) at the transmitter. Nonetheless, the optimum joint decoding and equalization can be overwhelmingly expensive. An suboptimum alternative known as turbo equalization has been, however, proven to be very efficient and effective.

The turbo concept was originated from the turbo decoding of concatenated error control codes [2]. When applied to equalization of coded communication systems, it exploits the synergy from coding and ISI on symbols and combines the soft (probabilistic) equalization and decoding in a turbo fashion to achieve greater performance improvement. The implementation of turbo detection separates the decoding and detection into commonly two or possibly more concatenated units, and the \textit{a posteriori} probability (APP) of symbols are calculated at each unit with the APP obtained at other units as the \textit{a priori} information. The processes then iterate among the units with judicious interleaving in between until the convergence of the APP.

Many low complexity soft equalization algorithms have been proposed in the literature for turbo equalization and among them the variations based on soft LMMSE equalizer (SLMMSE) have enjoyed the most success. Some new results in this respect can be found in [3]. Recently, an equalization algorithm based on a novel idea of probabilistic data association (PDA) was proposed in [4] and shown to provide better performance than the LMMSE equalizer. The basic idea of the PDA is to approximate the Gaussian mixture with a single Gaussian distribution, a concept seemingly simple yet very effective in practice. Another appealing advantage of the PDA is that it is soft in nature and thus applicable to turbo equalization.

In this paper, we investigate the turbo equalization using PDA. We first propose a general structure for PDA, which consists of a linear interference cancellation step followed by a probabilistic data association step in every iteration. Under the general structure, we first show that the original PDA belongs to one variation and it is computational inefficient. We then unveil that the popular SLMMSE equalizer can be considered as one sweep within a generalized PDA. Such connection implies that further performance improvement over the SLMMSE equalizer is possible if the PDA is applied instead in turbo equalization. We then demonstrate a way to incorporate the \textit{a priori} probability in the PDA algorithm, which makes PDA readily applicable to turbo equalization. We also provide the performance of the turbo equalization using PDA by simulations.

The remaining of the paper is organized as follows. The problem of equalization is formulated in Section II. The original PDA and the generalized PDA are discussed in Section III. Turbo equalization using the PDA is described in Section IV. In Section V, simulation results are presented. Finally, conclusions are drawn in Section VI.

II. PROBLEM FORMULATION

Consider a block coded communication system over an ISI channel. At the transmitter, the binary information data \{a\} are convolutionally encoded to obtain \{b\} at a code rate \(R\). The encoded symbols are then grouped into blocks of size \(N\) and the \(n\)th transmitted block is defined as \(b^n = [b(nN), \ldots, b(nN + N - 1)]^\top\). At the output of the channel coder, a code-bit interleaver is used to reduce the influence...
of the burst error. A zero-padding precoder is then used to insert guarding bits between successive data blocks, where a guarding-inserting matrix $P = \begin{bmatrix} I_N & 0 \end{bmatrix}_T^{(K-N)\times N}$ is first set by choosing $K \geq N + L$ and $I_N$ is an $N \times N$ identity matrix and then based on $P$, a block of $K$ symbol $s^n = Pb^n$ is produced. Each interleaved and precoded block $s^n$ is then BPSK modulated and transmitted through an ISI channel. Now, denote $h = [h(0), \cdots, h(L)]^T$ as the symbol-spaced tap coefficients in the channel response. The received data block $y^n$ can be represented by

$$y^n = Hb^n + n^n$$

where $H \in \mathbb{R}^{K \times N}$ is a tall Toeplitz matrix made from $h$ [4],

$$H = \begin{pmatrix}
    h(0) & 0 & 0 & \cdots & 0 \\
    h(1) & h(0) & 0 & \cdots & 0 \\
    \vdots & \vdots & \vdots & \ddots & \vdots \\
    h(L) & h(L-1) & h(L-2) & \cdots & 0 \\
    0 & \cdots & \cdots & \cdots & h(L) \\
    0 & 0 & 0 & \cdots & h(L-1) \\
    0 & 0 & 0 & \cdots & 0
\end{pmatrix}$$

$n^n$ is the white Gaussian noise vector with zero mean and covariance matrix $R_n = \sigma^2 I, \sigma^2 = N_0/2$ and $N_0$ is one-sided Gaussian power spectral density. In this model, ISI only exists inside each transmitted block, and does not propagate to adjacent blocks. Based on the received signal block $y^n$, our objective is to detect $b^n$, i.e., perform joint decoding and equalization.

For convenience, in the latter sections, we will ignore the superscript $n$ in the expression for $y,n$ and $b$.

**III. PDA E Qualization**

We first discuss the PDA equalization on uncoded systems.

**A. Original PDA Equalizer (PDAE)**

The PDA, originally proposed in [5] and later used for equalization in [4], is an iterative soft detection scheme that produces probabilistic information of data symbols at its output. Rather than calculate the soft information of all users at once, the PDA updates for each symbol in turns. The goal of PDA equalizer is to obtain the APP $p(b_i|y) \forall i$. Since

$$p(b_i|y) = \frac{p(y|b_i)p(b_i)}{\sum_{b_i} p(y|b_i)p(b_i)}$$

$p(b_i|y)$ can be obtained easily if $p(y|b_i)$ is known. However, to obtain $p(y|b_i)$, marginalization must be performed as

$$p(y|b_i) = \sum_{b_{-i}} p(y|b)p(b_{-i})$$

where $b_{-i}$ represents a $(N-1)\times 1$ signal vector that contains all data symbols except $b_i$. Apparently, (3) is of complexity exponentially increasing with the block length $N$ and thus computationally infeasible for large $N$. Note that the complexity is due to (3) being a mixture of Gaussian distributions. One simplification can be achieved by approximating the mixture by a single Gaussian, which is the essence of data association. This approximation is possible if we pretend the binomial random variables $b_{-i}$ as the Gaussians with mean and variance matched to the original binomials. Nonetheless, such approximation is still quite rough. To refine it, the PDAE employs an iterative scheme.

In detail, in the original PDAE, a zero-forcing (ZF) filter is first applied to the received block sequence $y$ to obtain

$$\tilde{y} = H^+ y = b + H^+ n$$

where $H^+$ is the Moore-Penrose pseudo-inverse of the channel matrix $H$. The filter output $\tilde{y}$ can be rewritten as

$$\tilde{y} = b_i e_i + \sum_{j=1,j \neq i}^{N} b_j e_j + \tilde{n}$$

where $b_i$ is the $i$th element of $b$, $e_i$ is the $i$th column of the $N \times N$ identity matrix, and $\tilde{n} = H^+ n$ is a $N \times 1$ Gaussian vector with zero mean and covariance matrix $R_{\tilde{n}} = H^+ R_n (H^+)^\top = \sigma^2 (H^+ H)^{-1}$. Define $\eta_i = P(b_i = 1|\tilde{y})$. Now, to avoid direct evaluation of $\eta_i$ from (5), which has a computational cost exponential in $N$, $b_i \forall j \neq i$ is approximated by a Gaussian random variable with mean and variance matched to its true ones. As a result, $\sum_{j=1,j \neq i}^{N} b_j e_j + \tilde{n}$ becomes a vector of Gaussian random variables, i.e.,

$$\sum_{j=1,j \neq i}^{N} b_j e_j + \tilde{n} \approx N(\mu, \Sigma)$$

where $\mu = \sum_{j \neq i} e_j \tilde{b}_j$ and

$$\Sigma = \sum_{j \neq i} e_j e_j^\top \sigma_{b_j}^2 + \sigma^2 (H^+ H)^{-1} = H^+ Q_i (H^+)^\top$$

and

$$\lambda(b_i) = \ln \frac{P(\tilde{y}|b_i = 1)}{P(\tilde{y}|b_i = -1)} = \exp \{2 e_i^\top \Sigma^{-1} (\tilde{y} - \mu)\}$$

Next, given a noninformative prior, the APP $\eta_i$, is approximated by

$$\eta_i \approx \frac{1}{1 + e^{-\lambda(b_i)}} \approx \frac{1}{2} \left\{1 + \tanh \frac{1}{2} \lambda(b_i)\right\}$$

Since (9) is only an approximation to $\eta_i$, the PDAE will, after one sweep of update from $i = 1$ to $N$, start another sweep until the APPs converge. In the end, the detection is performed as

$$\tilde{b}_i = \begin{cases} 1 & \eta_i \geq 0.5 \\ -1 & \eta_i < 0.5 \end{cases}$$
The above procedure is summarized in the following chart.

### The original PDAE

1. **Initialization:** Set $\eta_i = 0.5 \forall i$.
2. **Zero forcing:** Calculate $\tilde{y}_i = H^+ y$.
3. For $i = 1$ to $N$:
   - **Data association:** Compute $\mu$ and $\Sigma$ from (6) and (7).
   - **Probability update:** Calculate the LLR $\lambda(b_i)$ and $\eta_i$ according to (8) and (9).
4. **Convergence testing:** If the APPs converge, go to 5. Otherwise, go back to 3.
5. **Detection:** Detect $b_i$ according to (10).

Further computational reduction is possible and discussed in [5].

#### B. Another Look at PDA Equalizer

Notice that in the original PDAE, besides a zero forcing filter, any other linear filter can be used instead and different filters can be also applied at different $i$. Motivated by the observation, we formulate a general framework for the PDAE in this section.

In almost the same fashion as in the original PDAE, given some estimates about $\eta_i \forall i$, we first apply a general linear filter $W_i$ to $y$ to suppress the interference, i.e.,

$$
\bar{y}_i = W_i y = W_i (Hb + n_i) = W_i H \tilde{e}_i b_i + W_i (\sum_{j \neq i} H \tilde{e}_j b_j + n_i) = W_i H \tilde{e}_i b_i + \tilde{n}_i
$$

(11)

where $\tilde{n}_i = W_i (\sum_{j \neq i} H \tilde{e}_j b_j + n_i)$. We then perform the data association by approximating $\tilde{n}_i$ by a multivariate Gaussian distribution with the matching mean and covariance matrix, i.e., $\tilde{n}_i \approx N(\mu, \Sigma)$ with

$$
\mu = W_i (\sum_{j \neq i} H \tilde{e}_j b_j) = W_i m_i
$$

and

$$
\Sigma = W_i (\sum_{j \neq i} H \tilde{e}_j \tilde{e}_j^T \sigma_{b_j}^2 + \sigma^2 I) W_i^T = W_i Q_i W_i^T
$$

(13)

The corresponding log-likelihood ratio (LLR) is thus calculated by

$$
\lambda(b_i) = \ln \frac{p(\tilde{y}_i | b_i = 1)}{p(\tilde{y}_i | b_i = -1)} = 2(\bar{y}_i - \mu)^T \Sigma^{-1} W_i H \tilde{e}_i = 2(y - m_i)^T W_i^T (W_i Q_i W_i^T)^{-1} W_i H \tilde{e}_i
$$

(14)

With the above LLR, the $\eta_i$ and final hard decision can be calculated similarly according to (9) and (10). This generalized PDAE is summarized in the following chart.

#### The generalized PDAE

1. **Initialization:** Set $\eta_i = 0.5 \forall i$.
2. For $i = 1$ to $N$:
   - **Interference cancellation:** Calculate $\tilde{y}_i = W_i y$.
   - **Data association:** Compute $\mu$ and $\Sigma$ from (12) and (13).
   - **Probability update:** Calculate the LLR $\lambda(b_i)$ and $\eta_i$ according to (14) and (9).
3. **Convergence testing:** If the APPs converge, go to 4. Otherwise, go back to 2.
4. **Detection:** Detect $b_i$ according to (10).

Under the above general framework, different choices on $W_i$ will lead to different algorithm implementations. The original PDAE corresponds to the case for $W_i = (H^T H)^{-1} H^T$, i.e., a zero forcing filter is applied to the interference cancellation. We will discuss next some other choices on $W_i$ and reveal some important properties.

1. **Case I:** $W_i$ is invertible: In this case, the LLR is calculated by

$$
\lambda^1(b_i) = 2(y - m_i)^T W_i^T (W_i Q_i W_i^T)^{-1} W_i H \tilde{e}_i = 2(y - m_i)^T Q_i^{-1} H \tilde{e}_i
$$

(15)

Notice that no $W_i$ appears in the calculation of $\lambda(b_i)$ in (15). Since our objective is to obtain $\eta_i$s and the LLRs are all that we need to calculate, this important observation implies that the use of $W_i$s, i.e., the interference cancellation steps are redundant. No matter what the special linear filter $W_i$s are, the resulting PADEs will all be equivalent to $W_i = I$, i.e., no interference cancellation step is needed.

2. **Case II:** $W_i = \tilde{W}_i H^T$, where $\tilde{W}_i$ is an invertible matrix: The scenarios that fall into this case include $W_i = \tilde{W}_i H^T$, $W_i = H^+$ (the zero forcing filter), and $W_i = (H^T H + \sigma^2 I)^{-1} H^T$ (the linear MMSE filter). In this case, the LLR is calculated by

$$
\lambda^2(b_i) = 2(y - m_i)^T \tilde{W}_i H \tilde{W}_i^T (W_i H^T Q_i W_i^T)^{-1} W_i H^T \tilde{e}_i = 2(y - m_i)^T (H^T Q_i H)^{-1} H^T \tilde{e}_i
$$

(16)

We observe that $\tilde{W}_i$ is not needed for the LLR and thus the presence of $\tilde{W}_i$ in the linear filter $W_i$ is unnecessary. Since the original PDAE belongs to this case, we conclude that its implementation is inefficient. The proper implementation is to apply simply $W_i = H^T$ at interference cancellation.

#### C. Connection between the PDA and Soft LMMSE equalizers

The objective of the popular SLMMSEE [3] is to calculate the LLR $\lambda^{\text{slmmsee}}(b_i)$ for $i = 1, \ldots, N$. Based on a set of the *a priori* probabilities $P(b_i)$, which is assumed for the moment to be $\eta_i \forall i$, the LLR can be shown as [3]

$$
\lambda^{\text{slmmsee}}(b_i) = \frac{2(y - \tilde{H} \tilde{b} + \tilde{b} \tilde{e}_i)^T (c_i^{-1} \tilde{b}^{-1} H \tilde{e}_i) - 1}{e_i^T H^T c_i^{-1} H \tilde{e}_i}
$$

(17)
where \( \bar{b}_i \) is the mean of \( b_i \) calculated from \( \eta_i \), \( \bar{b} = [\bar{b}_1, \ldots, \bar{b}_n]^\top \), \( c_i = Q_i + He_i e_i^\top H^\top \), and \( m_i \) and \( Q_i \) are defined as in (12) and (13). Now, given the same \( \eta_i \) being an estimate APP for \( b_i \), let us revisit the LLR (15) and try to express it in terms of \( c_i \).

\[
\lambda^1(b_i) = 2(y - m_i)^\top Q_i^{-1} H e_i = 2(y - m_i)^\top (c_i - He_i e_i^\top H^\top)^{-1} H e_i = 2(y - m_i)^\top (c_i^{-1} + \frac{c_i^{-1} He_i e_i^\top H c_i^{-1}}{1 - e_i^\top H c_i^{-1} H e_i}) e_i = 2(y - m_i)^\top c_i^{-1} He_i (1 + \frac{e_i^\top H c_i^{-1} H e_i}{1 - e_i^\top H c_i^{-1} H e_i}) = \frac{2(y - m_i)^\top c_i^{-1} H e_i}{1 - e_i^\top H c_i^{-1} H e_i}
\]

where the third equality is the result of the WoodBury’s identity. Surprisingly, (18) is the same as (17). This similarity suggests that the popular SLMMSE equalizer is equivalent to one sweep within the PDAE of case I, i.e., the case where no interference step is needed. Nonetheless, from a generalized PDA perspective, the APPs of the data bits may not converge at one iteration. As a result, this connection implies that further performance improvement over the SLMMSE equalizer is possible if the PDAE is applied instead in a turbo equalization.

IV. TURBO EQUALIZER USING PDAE

A. Basic Principle of Turbo Equalization

Turbo equalization scheme is proposed in [2] for convolutionally coded digital transmission over intersymbol interference channels [3], [6], [7]. Turbo equalization provides the near optimal solutions for joint equalization and decoding at manageable cost. The structure of the turbo equalization receiver is shown in Figure 1. The turbo equalizer is an iterative detection scheme that consists of two units: a soft-input soft-output (SISO) equalizer and a SISO decoder. The extrinsic information of the data bits are exchanged back and forth between the two units in a turbo fashion until the convergence of the APPs. The output of the turbo equalization is then a near optimal approximation of the true APPs.

In detail, given the received observation \( y \), the SISO equalizer computes the APPs, \( P(b_i|y) \), and outputs a posteriori LLR \( \Lambda_1(b_i) \) by

\[
\Lambda_1(b_i) = \ln \frac{P(b_i = 1|y)}{P(b_i = -1|y)} = \ln \frac{P(y|b_i = 1)}{P(y|b_i = -1)} + \ln \frac{P(b_i = 1)}{P(b_i = -1)} = \Lambda_1(b_i) + \lambda^2_2(b_i)
\]

where the first term \( \lambda_1(b_i) \) is the LLR, which is often referred to as the extrinsic information and the second term \( \lambda^2_2(b_i) \) represents the log-prior ratio, which is called the intrinsic information. The intrinsic information at the SISO equalizer is the extrinsic information obtained by SISO decoder in the previous iteration. For the first iteration, when no prior information is available, \( \lambda^2_2(b_i) = 0 \) for all \( i \). At the output of the SISO equalizer, the data bits are deinterleaved and the corresponding extrinsic information \( \lambda_1(b_i) \) is then fed into the SISO decoder of code bit \( i \) as the a priori or the intrinsic information of the code bit.

At the SISO decoder, based on the extrinsic information from the equalizer \( \lambda_1(b_i) \) and the trellis structure of the channel code, the SISO decoder computes its a posteriori LLR \( \Lambda_2(b_i) \) by

\[
\Lambda_2(b_i) = \lambda_2(b_i) + \lambda^2_2(b_i)
\]

where \( \lambda^2_2(b_i) = \lambda_1(b_i) \). Afterwards, the extrinsic information \( \lambda_2(b_i) \) at the decoder is then interleaved and fed back to the equalizer as the intrinsic information of the code bits in the next iteration. After a few iterations, \( \Lambda_1(b_i) \) and \( \Lambda_2(b_i) \) will be very close to the true a posteriori LLR.

B. Turbo Equalization using PDAE

The PDAE can be used in a turbo equalizer to provide the soft information \( \lambda_1(b_i) \) for the SISO channel decoder. However, the PDAEs discussed in the section III only calculate the LLR and do not yet accept the prior information at their input. To develop a SISO PDAE which incorporates the prior information, we observe from (2) that given an estimate of the LLR, the APP \( \eta_i \) is updated by

\[
\Lambda_1(b_i) = \ln \frac{\eta_i}{1 - \eta_i} \approx \lambda(b_i) + \lambda^2_2(b_i)
\]

where \( \lambda(b_i) \) is an approximation to the extrinsic information \( \lambda_1(b_i) \) and under the general framework is calculated by (14), and \( \lambda^2_2(b_i) \) is the a priori LLR, which, in a turbo detection, is provided by the SISO decoder. Note now that in the SISO PDAE, the estimate of the APP \( \eta_i \) should be calculated based on \( \Lambda_1(b_i) \) but the LLR \( \lambda(b_i) \) as before. However, when \( \lambda^2_2(b_i) = 0 \), which is often assumed at the very beginning of the turbo equalization, the SISO PDAE is exactly the same as the PDAEs discussed in section III.

We want to, however, point out that the PDAE is not an optimal equalizer and there is a possibility that the PDAE finds a non-optimal solution. Such problems can happen especially at early iterations of a turbo equalizer. To overcome the problem, less PDA iterations would be allowed at early turbo iterations.
A more elegant solution will be to adjust adaptively the maximum number of the PDA iterations according to an estimate of the current signal to noise ratio.

V. SIMULATION

In this section we illustrate the performance of turbo equalization using the PDAE by comparing with that of the exact SLMMSEE proposed in [3]. We consider a block coded transmission system, in which rate $R = 1/2$ convolutional code with generators [7,5] in octal notation and block size $N = 512$ is employed. We select the length-3 ISI channel, with the coefficients $h_0 = 0.407, h_1 = 0.815, h_3 = 0.407$ and a random interleaver for each block. We consider only the BER-optimal MAP approach implemented by the BCJR algorithm as the SISO channel decoder.

In Figures 2-4, the BER vs. $E_b/N_0$ curves under different numbers of iterations are used to compare the performance of turbo equalization between PDAE and exact SLMMSEE, where $E_b$ is the information bit energy. The lower bound on the BER performance is that of coded data transmission over an AWGN channel at the same $E_b/N_0$.

Figure 2 shows the performance of separate or one-time equalization and decoding. As expected, the PDAE exhibits better performance than the exact SLMMSEE. The similar trend is observed in Figures 3 and 4, as the iteration number increases.

VI. CONCLUSION

We investigated in the paper turbo equalization by PDA. We proposed a general structure for the PDAE, based on which several algorithm variations have been derived. From these variations, we observe that the original PDA is computational inefficient. We also drew that connection between that the PDAE and the popular SLMMSEE and indicated that the PDAE could provide better performance than the SLMMSEE when applied in turbo equalization. To use the PDAE in turbo equalization, we showed the way to incorporate the a priori information in PDAE. The simulation results demonstrated good performance of turbo equalization using PDA.

REFERENCES

[1] J. G. Proakis, Digital Communication, 3rd ed. New York: McGraw-Hill, 1995.
[2] C. Douillard, “Iterative correction of intersymbol interference: Turbo Equalization,” Eur. Trans. Telecommun., vol. 6, pp. 507–511, Sept.-Oct. 1995.
[3] M. Tüchler, R. Koetter, and A. C. Singer, “Turbo equalization: Principles and New Results,” IEEE Transactions on Communications, vol. 50, no. 5, pp. 754–767, May 2002.
[4] Z. Tian, “Soft decision feedback equalization for pre-coded FIR-MIMO systems,” in Proceedings of IEEE Sensor Array and Multichannel Signal Processing Workshop, Aug. 2002, pp. 594–598.
[5] J. Luo, K. R. Pattipati, P. K. Willett, and F. Hasegawa, “Near-optimal multiuser detection in synchronous cdma using probabilistic data association,” IEEE Communication Letters, vol. 5, no. 9, pp. 361–363, Sept. 2001.
[6] X. Wang and H. V. Poor, “Iterative(turbo) soft interference cancellation and decoding for coded CDMA,” IEEE Transactions on Communications, vol. 47, pp. 1046–1061, July 1999.
[7] P. H. Tan, L. K. Rasmussen, and J. Luo, “Iterative multiuser decoding based on probabilistic data association,” in Proceedings of IEEE International Symposium on Information Theory, June 2003, p. 301.