Abstract—We present a simple, effective iterative structure for joint source-channel decoding using global iterations between a source and channel decoder. The source decoder uses the Sequence Based Approximate Minimum Mean Square Error (SAMMSE) algorithm. The iterative structure is tested using both LDPC and turbo channel codes and simulations show significant performance gains over non-iterative systems. To avoid transmitting the parameters required by the SAMMSE decoder, we estimate them within the iterative structure.

I. INTRODUCTION

Since the discovery of turbo codes [1], iterative decoding has attracted widespread use in communication systems. The turbo principle [2] involves finding effective ways to interface different soft-in/soft-out (SISO) modules. According to Shannon [3], source and channel coding can be treated separately, the premise being that both the source and channel codes are "perfect". This means the source encoder completely removes all redundancy from the information and the channel code gives zero error probability above a certain $E_b/N_0$. In practice however, there is always some residual redundancy at the source encoder's output which can be exploited at the decoder, thereby allowing a joint source-channel (JSC) structure ([4], [5], [6], [7]).

The use of iterations between a source and channel decoder has been investigated in [8] where Peng et al propose an iterative decoder where the channel decoder is a conventional turbo decoder and the source decoding algorithm differs according to the type of compression used. Based on information from the source decoder, Peng et al scale the turbo decoder soft outputs, to increase their reliability. The scaling factor and the number of iterations are selected empirically. Up to 0.6 dB reduction in $E_b/N_0$ has been achieved. In [9], Frias and Villasenor propose joint turbo decoding and estimation of hidden Markov sources where a binary source is modelled as a Markov source. The trellis describing the Markov source is treated as a source decoder, which then exchanges soft information with each of the constituent turbo decoders, giving a gain of about 1 dB. [9] introduces the iterative estimation of the source parameters during decoding using the Baum-Welch algorithm [10]. [11] proposes iterative source-channel decoding for image transmission, where the source redundancy is modelled by a Markov random field (MRF). The decoder is similar to a turbo decoder, where the inner decoder is the SISO channel decoder, but the outer decoder is the source decoder. The SAMMSE decoder was introduced by Miller and Park in [12] and reviewed in Section III of this paper. [6] used the SAMMSE decoder along with a binary convolutional code, but did not iterate between the source and channel decoders.

Here, we build on ideas from [13] and introduce a simple and effective technique to achieve JSC decoding. We introduce global/joint iterations between the SAMMSE decoder and the channel decoder. Traditionally, convolutional and turbo codes are considered in JSC decoders, but our method is applicable to any SISO channel decoder. Our decoder has been tested using both LDPC ([14], [15], [16]) and turbo codes. Simulations show JSC decoding gives a 0.4 to 1.5 dB reduction in $E_b/N_0$ with fewer iterations. Similar to [9], we adapt the Baum-Welch algorithm to estimate (at the receiver) the source parameters for the SAMMSE decoder. Finally, when LDPC codes are used, we propose retaining soft information within the channel decoder between joint iterations, which gives additional gains in the system performance. Hence, while the concept of iterative JSC decoding is not new, the main contributions of this paper are the development of a SISO structure using the SAMMSE decoder and adapting a known parameter estimation algorithm within the iterative framework. This allows us to use known codes in a high performance, flexible JSC decoder and apply (in principle) established methods to predict convergence ([17], [18], [19]).
II. SYSTEM MODEL

A. Transmitter

The transmitter model is described in Fig. 1. The redundancy between the source symbols is modelled using a first order Gauss-Markov source with correlation coefficient \( \rho \). The symbols generated by the source are \( \mathbf{s} = (s_0, s_1, ..., s_{T-1}) \) which are passed to a uniform scalar-quantizer with \( M \) quantization levels and hence, \( M \) possible indices \{0, 1, ..., \( M-1 \)\}. Corresponding to \( \mathbf{s} \), the transmitted indices are \( \mathbf{l} = (I_0 = i_0, I_1 = i_1, ..., I_{T-1} = i_{T-1}) \). Each index is mapped (natural mapping) to \( \log_2(M) \) bits, giving a bit sequence \( \mathbf{b} = (b_0, b_1, ..., b_T \log_2 M - 1) \), which are passed to the channel encoder. The code sequence is \( \mathbf{c} = (c_0, c_1, ..., c_{N-1}) \). The output from the channel encoder is BPSK modulated, (denoted by \( \chi \)) and transmitted through an AWGN channel. We would like to point out that even though no variable-length (lossless) source encoding was used, our decoder would still give a performance gain if this were used because typically there is always residual redundancy at the source encoder’s output [7]. This redundancy is exploited at the decoder to obtain improved performance. We model the transmitted indices as the states of a Markov model. This allows us, at the decoder side, to implement source decoding inference and parameter estimation algorithms based on hidden Markov models.

B. Receiver

Our model for the receiver is shown in Fig. 2. Our proposed decoder has two stages, a channel decoder and a source decoder, where soft information is exchanged between the two decoders. This iterative process is indicated in Fig. 2 by the dotted lines showing two-way information exchange between the two decoders. There are two levels at which iterations are carried out. At the first level, iterations are carried out within the channel decoder, giving soft outputs \( L(Q) \), used to estimate bit probabilities. From these estimates we obtain symbol probabilities required by the SAMMSE decoder. During parameter estimation, the Baum-Welch algorithm is implemented to estimate the initial index probabilities i.e. \( P[I_0 = l], l = 0, 1, ..., M - 1 \) and the conditional probabilities \( P[I_t = l|I_{t-1} = k], l, k = 0, 1, ..., M - 1 \) which are also required by the SAMMSE algorithm. \( \mathbf{j} = (J_0 = j_0, J_1 = j_1, ..., J_{T-1} = j_{T-1}) \) are the “noisy” indices at the source decoder input. Unlike [12] however, we do not get \( \mathbf{j} \) directly from the channel due to the presence of the channel decoder. The source decoder produces a posteriori (AP) index probabilities which are then mapped to a priori bit probabilities \( L(\mathbf{c}) \) for the channel decoder. Hence, at the second level, we have the global iterations where we pass information between our two decoders. Our overall iterative decoding scheme differs from a conventional receiver where a fixed number of channel decoder iterations are performed followed by source decoding, i.e. a receiver in Fig. 2 without the dotted lines.

III. PROPOSED ITERATIVE DECODER

A. SAMMSE Decoder

The sequence-based approximate MMSE decoder was introduced in [12] for source decoding over noisy memoryless channels. In this, the authors exploit the redundancy between the transmitted symbols which often remains after source encoding. [12] develops a decoder that minimizes a (close) approximation of the end-to-end MSE (between the source and reconstruction). The quantized source and channel tandem is modelled as a hidden Markov model where, from the decoder’s perspective, the hidden states correspond to the transmitted indices. It is shown that this decoder outperforms previously proposed schemes for decoding of sources with memory over noisy channels. The distortion in \( \mathbf{s} \), given reconstructed symbols \( \mathbf{\hat{s}} \) is:

\[
D(\mathbf{s}, \mathbf{\hat{s}}) = \sum_{t=0}^{T-1} d(s_t, \hat{s}_t)
\]  
(1)

Where \( d(s_t, \hat{s}_t) = |s_t - \hat{s}_t|^2 \). The cost function to be minimized is the approximate expected distortion given \( J \):

\[
\hat{D} = \sum_{t=0}^{T-1} \sum_{l=0}^{M-1} |E[s_t|I_t = l] - \hat{s}_t|^2 P[I_t = l|J = j]
\]  
(2)

\( P[I_t = l|J = j] \) is found using the following forward/backward algorithm,

1) Initialize the forward recursion variable:

\[
\alpha_0[l] = P[I_0 = l] P[J_0 = j|I_0 = l], l, j = 0, ..., M - 1
\]

2) For \( l, j = 0, 1, ..., M - 1 \ t = 1, 2, ..., T - 1 \), compute:

\[
\alpha_t[l] = \sum_{k=0}^{M-1} \alpha_{t-1}[k] P[I_t = l|I_{t-1} = k]
\]  
(3)

\( P[J_t = j|I_t = l] \)

3) Initialize the backward recursion variable:

\[
\beta_{T-1}[l] = 1, l = 0, ..., M - 1
\]

4) For \( l, j = 0, 1, ..., M - 1 \ t = T - 2, ..., 0 \), compute:

\[
\beta_t[l] = \sum_{k=0}^{M-1} P[I_{t+1} = k|I_t = l] P[J_{t+1} = j|I_{t+1} = k] \beta_{t+1}[k]
\]  
(4)

\( \beta_t[l] \)
5) The a posteriori symbol probability is computed as:
\[
P[I_t = l | J = j] = \frac{\alpha_t[l] \beta_t[j]}{\sum_{m=0}^{N-1} \alpha_t[m] \beta_t[m]} \tag{3}
\]

6) The reconstructed symbol is:
\[
\hat{s}_t = \sum_{l=0}^{M-1} E[s_t | I_t = l] P[I_t = l | J = j] \quad \forall t \tag{4}
\]

Since the working of turbo and LDPC codes are very well known and documented, we have omitted details about their working. Besides the seminal papers, some useful references can also be found in [20], [21], [22], [23].

B. Joint source-channel decoding with global iterations

The channel decoder receives the corrupted signal, \( y \), from the channel. After performing channel decoder iterations, the soft output \( L(Q) \) is obtained. Using these, the \( AP \) bit probabilities \( P[c_r = 0 | y] \) and \( P[c_r = 1 | y] \) are found. Previously, we saw that the \( \text{SAMMSE} \) decoder requires the transition probabilities, \( P[J_t = j | I_t = l] \). The \( \text{SAMMSE} \) decoder no longer receives the noisy indices \( J \) from the channel, so instead, we propose using the \( AP \) bit probabilities from our channel decoder to assist the source decoder. This is done by estimating \( P[J_t = j | I_t = l] \) from bit probabilities, under a conditional independence assumption of the bit probabilities. This can be written as:
\[
P[J_t = j | y, I_t = l] = \prod_{i=1}^{\log_2 M} P[c_{r+i-1}(l) | y] \tag{5}
\]

where \{\( c_r(l) \), ...\( c_{r+\log_2 M-1}(l) \)\} denote the bits corresponding to the symbol \( I_t = l \), i.e. if \( I_t = 0 \), then \( c_r(l) = 0 \) and \( c_{r+1}(l) = 0 \) (with \( M = 4 \)). With these approximations, source decoding is carried out. The \( \text{SAMMSE} \) decoder produces the \( AP \) symbol probabilities \( P[I_t = l | J = j] \) (3).

We use these to find better estimates of the a priori bit probabilities by summing them over symbols that have the same bit values. These are passed to the channel decoder, thereby achieving a global iteration. This iterative process can be repeated until convergence, or a maximum number of global iterations have been reached.

The soft information from the channel decoder can also be used in the Baum-Welch algorithm (BW) to estimate \( P[I_0 = l], l = 0, 1, ..., M-1 \) and the conditional probabilities \( P[I_t = l | I_{t-1} = k], l, k = 0, 1, ..., M-1 \). Details on the algorithm can be found in [24]. Briefly, a hidden Markov model is characterized by \( \lambda = A, B, \pi \) where \( A \) is the state (index) transition probability, \( A = P[S_j | S_l] \) for states \( 0 \leq S_j, S_l \leq M - 1 \). \( \pi \) is the initial state distribution and \( B = P[J_t = j | S_l] = b^y_j(J_t) \) for a given observation \( J_t = j \) and state \( S_l \). If one is directly given the observation sequence \( J = j \), then BW can also be used to estimate \( B \).

However, since the channel decoder does not explicitly give us \( j \), but rather bit a posteriori probabilities, it is difficult to estimate \( B \) within the BW framework. Instead, we form an “instantaneous” estimate of the observation probability mass function, conditioned on a particular state, again applying a conditional independence assumption:
\[
\begin{align*}
b_0(J_t) &= P[c_r = 0 | y] P[c_{r+1} = 0 | y] \\
b_1(J_t) &= P[c_r = 0 | y] P[c_{r+1} = 1 | y] \\
b_2(J_t) &= P[c_r = 1 | y] P[c_{r+1} = 0 | y] \\
b_3(J_t) &= P[c_r = 1 | y] P[c_{r+1} = 1 | y] \tag{6}
\end{align*}
\]

Here we are assuming there are four states in Markov model and \{\( c_r, c_{r+1} \)\} denote bits corresponding to symbol time \( t \). Note that the instantaneous estimates (6) implicitly assume a time-varying \( B \). Given this estimate of \( B \), the BW algorithm updates \( A \) and \( \pi \) with each global iteration.

<table>
<thead>
<tr>
<th>Channel Code</th>
<th>No JSC</th>
<th>JSC</th>
<th>Total iter.</th>
</tr>
</thead>
<tbody>
<tr>
<td>ExIRA</td>
<td>0.95</td>
<td>-0.45 (-0.5981)</td>
<td>40</td>
</tr>
<tr>
<td>DVB-S2</td>
<td>0.39</td>
<td>-0.0192 (-0.49)</td>
<td>50</td>
</tr>
<tr>
<td>CDMA00, rt.1/5</td>
<td>0.058</td>
<td>-0.3147</td>
<td>50</td>
</tr>
<tr>
<td>CDMA00, rt.1/3</td>
<td>0.508</td>
<td>-0.68</td>
<td>60</td>
</tr>
</tbody>
</table>

IV. SIMULATION RESULTS

Extensive simulations were carried out to investigate the performance of the JSC decoder. We use a Gauss-Markov source with \( \rho = 0.9 \) and uniform scalar quantizer with \( M = 4 \). Three different channel codes are considered, the Extended Irregular Repeat Accumulate (ExIRA, rate 1/3, data length 5000 bits) LDPC code [25], the irregular DVB-S2 LDPC code (rate 1/3, length 5400) [23] and CDMA2000 turbo code (rate 1/5, length 4602 and rate 1/3, length 12282) [26]. Fig. 3-6 and Table 1 show a gain of 0.4 – 1.5 dB in \( E_b/N_0 \) with global iterations for the three channel codes (No JSC implies 100 channel decoder and 0 global iterations). When using LDPC codes, we have the option of retaining the soft information at the check and variable nodes.
of the LDPC decoder, between global iterations per frame, giving an additional 0.15 – 0.4 dB gain in $E_b/N_0$ (shown in parenthesis in Table 1). JSC decoding also reduces the number of “total” iterations by almost half i.e. 10 DVB-S2 and 5 global iterations (50 total), give the same BER (at lower $E_b/N_0$) as 100 DVB-S2 iterations, without JSC decoding. [11] states that by using a MRF source instead of a Gauss-Markov source, a priori source probabilities need not be stored. Here, we overcome this storage requirement by estimating the probabilities during the JSC decoding. The effectiveness of our parameter estimation is shown in Fig. 7. Estimation is performed from scratch for each transmitted frame of data, though this may not be needed since the source parameters do not vary significantly from frame to frame. Fig. 8 shows the mutual information (MI) between reconstructed symbols (SAMMSE decoder) and source symbols (transmitter), with and without JSC decoding. MI increases with global iterations, suggesting that JSC decoding gives us progressively better estimates of source symbols. We also observed that more powerful codes gained less from JSC decoding.

V. Conclusions

A simple, effective JSC scheme has been proposed with the SAMMSE decoder and both, turbo and LDPC channel codes. No modification in the constituent decoder structures is necessary and the diverse selection of codes show the flexibility and robustness of our decoder. Gains between 0.4 – 1.5 dB are shown. JSC decoding increases the mutual information between the transmitted and reconstructed source symbols. Source parameters have also been estimated during global iterations. We found that density evolution methods based on a Gaussian approximation of extrinsic information ([17], [18]) were not a good convergence predictor for our decoder. We speculate this is because the information from the SAMMSE decoder looses its Gaussian distribution after a few global iterations. Hence the analysis in [19] may be more reliable for our decoder, this is a work in progress. Finally, an alternative approach to JSC decoding would be to consider a unified iterative receiver design based on factor graphs [27] and though [27] focuses on a joint, iterative demodulator-decoder, it may be possible to incorporate source decoding into such a receiver, as we can already exchange soft information between the decoders.

REFERENCES

Pb

Fig. 6. BER vs. $E_b/N_0$ with and without JSC decoding for rate 1/3 CDMA2000 code with data length of 12282 bits.

Fig. 7. BER vs. number of global iterations (with 10 LDPC iter.) at select $E_b/N_0$ for a rate 1/3 ExIRA code, data length 5000 bits. Solid lines indicate BER when source probabilities are known a priori at the receiver. Dotted lines illustrate BER performance when they are estimated during global iterations. Comparable results were also obtained using the other channel codes.

Fig. 8. Mutual information between the source symbols and reconstructed source symbols, using rate 1/3 ExIRA code (5000 bits). Comparable results were also obtained using the other channel codes.