Factor Graph Based Derivation of a Receiver for PCC Coded Transmissions over a Frequency Selective Fading Channel

M. F. Flanagan and A. D. Fagan
University College Dublin, Belfield, Dublin 4, mark.flanagan@ee.ucd.ie, tony.fagan@ucd.ie

Abstract

In previous work, we presented a receiver to detect and decode parallel concatenated convolutional (PCC) coded transmissions over a frequency selective Rayleigh fading channel. This scheme generalized work by Valenti and Woerner to frequency selective channels. It was shown that re-estimation of the channel after each PCC decoding iteration yields significant gain. In this paper we present a detailed derivation of our receiver from factor graph theory. This illustrates the interaction between the channel estimation procedure and the message passing algorithm. In particular it is shown that, in contrast to Valenti and Woerner's scheme, the pilot symbols appear as variables in the factor graph.

1 Introduction

The sum-product algorithm, which operates in a factor graph [3], may be used to design iterative receivers for finite-state transmitter-channel combinations [4]. It has also been shown to give rise to the best known classes of error correcting codes, namely PCC and LDPC codes. When the channel is unknown, it is desirable to incorporate channel estimation into the iterative algorithm in a manner which combines somewhat naturally with the sum-product algorithm. One method of achieving this for flat-fading channels is to approximate the channel as finite-state, and then simply apply the sum-product algorithm directly. However for a frequency selective channel, this method has high complexity as the number of channel states is exponential in the channel length.

In this paper we show that the channel estimation procedure of [1] combines with the sum-product algorithm as follows. First, the sum-product algorithm operates in the same way as if the channel parameters were known exactly, *i.e.* the state space of the L-tap channel is not expanded beyond 2^L. The effect of this is that a subset F of the pendant factor nodes in the graph require knowledge of the channel; channel estimates are substituted for the actual values. Then, after each iteration of message passing, summaries are computed for all binary variables, and these summaries are used to update the channel estimates. The subsequent iteration of message passing then begins with the re-sending of messages from all nodes in F with the updated channel estimates in place of the actual values.

2 System Model

This section describes the structure of the transmitter and presents our channel model, and also serves to introduce detailed mathematical notation to facilitate the derivation of section 3. We provide a block diagram of the iterative receiver and give an intuitive explanation of its operation, prior to its formal derivation in section 3.

2.1 Transmitter

Figure 1 shows the discrete-time model of the transmitter. The two blocks marked "RC encoder" in the figure are rate-1 recursive convolutional encoders. The information bit sequence $\{u_k\}, k = 1, 2, \ldots N$ is encoded by RC encoder $R1$ to produce the sequence of parity bits for the upper encoder, $\{a_k\}, k = 1, 2, \ldots N$. The state sequence followed by $R1$ is $\{s_k\}, k = 0, 1, \ldots N$. The information bit sequence is also interleaved by the interleaver Π to form $\{u'_k\}, k = 1, 2, \ldots N$, where the interleaving is to be interpreted as $u'_k = u_{\Pi(k)}$. The inverse interleaver is denoted Πinv, *i.e.* $u'_{\Pi inv(k)} = u_k$. The interleaved sequence is encoded by RC encoder $R2$ to produce the sequence of parity bits for the lower encoder, $\{b_k\}, k = 1, 2, \ldots N$. The state sequence followed by $R2$ is $\{s'_k\}, k = 0, 1, \ldots N$. The two parity bit sequences are then punctured. We denote all unpunctured index positions in the upper encoder's parity output by S_a, and all unpunctured index positions in the lower encoder's parity output by S_b. The index sets S_a and S_b are subsets of $\{1, 2, \ldots N\}$ of size N_a and N_b, respectively. The punctured parity sequences are then multiplexed with the information sequence itself to form the PCC codeword. The set of all unpunctured index positions in the sequence $\mathbf{u}$ is denoted S_u and is simply $\{1, 2, \ldots N\}$ as the PCC code is systematic. We assume that the trellis of the upper encoder starts and ends in the zero state, and that the trellis of the lower encoder starts in the zero state.

The codeword is interleaved by a block channel interleaver Π_C in order to combat burst errors caused

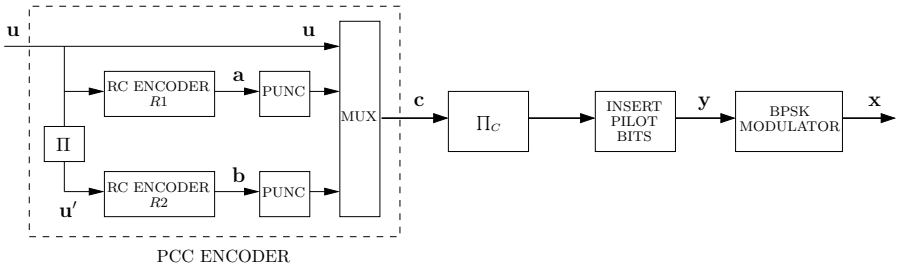

Fig. 1. Transmitter Architecture

by the fading channel. The resulting bit sequence is parsed into blocks of M consecutive bits (M is called the *pilot block spacing*) and a group of $2L-1$ known bits is inserted into the centre of each block (M is assumed even). Let $\{q_k\}, k = 1, 2, \ldots N_q$ denote the pilot bits in the order in which they are inserted into the bitstream, and let $S_q = \{1, 2, \ldots N_q\}$. Denote by $D(k)$ the (known) value of pilot bit q_k. We define the indicator function $J_k(q_k) = \text{Ind}\{q_k = D(k)\}$. Let $N_t = N + N_a + N_b + N_q$. The combination of PCC multiplexer, channel interleaver and pilot insertion process may be represented by the permutations $U : S_u \to S'_u$, $A : S_a \to S'_a$, $B : S_b \to S'_b$ and $Q : S_q \to S'_q$ such that

$$\begin{array}{llll} u_k = y_{U(k)} & k \in S_u & b_k = y_{B(k)} & k \in S_b \\ a_k = y_{A(k)} & k \in S_a & q_k = y_{Q(k)} & k \in S_q \end{array}$$

We note that $S'_u \cup S'_a \cup S'_b \cup S'_q = \{1, 2, \ldots N_t\}$. The resulting bit sequence $\{y_k\}, k = 1, 2, \ldots N_t$ is BPSK modulated to $\{x_k = \mathcal{M}(y_k)\}, k = 1, 2, \ldots N_t$. We adopt the convention $\mathcal{M}(0) = -1$, $\mathcal{M}(1) = +1$.

2.2 Channel

The frequency selective channel is modelled by a time-varying L-tap FIR filter followed by the addition of AWGN. The receive signal is given by

$$r_k = \sum_{j=0}^{L-1} h_k^{(j)} x_{k-j} + n_k = z_k + n_k \qquad (1)$$

The statistics of the $\left\{h_k^{(j)}\right\}$ are omitted as they are irrelevant to the derivation of section 3. The additive noise sequence $\{n_k\}$ is white, each element being a complex-valued Gaussian random variable with zero mean and variance σ^2 in each of the real and imaginary directions.

2.3 Receiver

A block diagram of the receiver architecture is shown in figure 2. This receiver shall be derived in detail using factor graph theory in section 3. Prior to the formal derivation of this block diagram, we give an overview of its operation.

The equalizer SISO module and PCC decoder SISO module exchange extrinsic information $\left\{\lambda_k^{(HR)}\right\}$ and $\left\{\lambda_k^{(RH)}\right\}$ on the transmit bit sequence, in the manner of Douillard's original scheme [5]. The difference here is that the information corresponding to pilot symbols is removed at the equalizer output as this is of no relevance to the decoder, and LLRs corresponding to the pilot bits are inserted in the feedback path. The inserted LLRs are ideally infinite in magnitude as they correspond to *a priori* information on the pilots.

The equalizer SISO requires estimates of the time-varying complex channel taps and the AWGN noise variance in order to form its branch metrics. These estimates are provided by a channel estimator the details of which are given in [1].

System performance is considerably improved by re-estimating the channel after each (equalizer SISO, decoder SISO) iteration. This is achieved by including the feedback path shown in dotted lines in figure 2. The summaries $\left\{L_k^{(y)}\right\}$ for the PCC code bits are interleaved and fed to a nonlinear function which transforms these LLRs into bit estimates. The original pilot symbols are re-inserted into the bit stream and the resulting estimated transmit stream $\{\hat{x}_k\}$ is used by the channel estimator to produce a refined channel estimate. This improved estimate aids the equalizer SISO in the next iteration.

3 Derivation of Receiver Algorithm

In this section we derive the receiver in detail using factor graph theory. This will illustrate the interaction between the sum-product algorithm and the channel estimation procedure. We will use the original factor graphs of [3], as opposed to the more compact "Forney graphs" of [6]. Also, instead of passing the message $f(x)$, we pass either the log domain message, $\log f(x)$, or (for $x \in GF(2)$) the LLR for the message, $\log(f(1)/f(0))$.

3.1 Preliminaries - Markov Finite State Machines

A Markov finite state machine (MFSM) is a system such that the current input and current state uniquely determine the current output and next state, *i.e.*

$$(u_k, s_{k-1}) \quad \text{uniquely determine} \quad (x_k, s_k) \qquad (2)$$

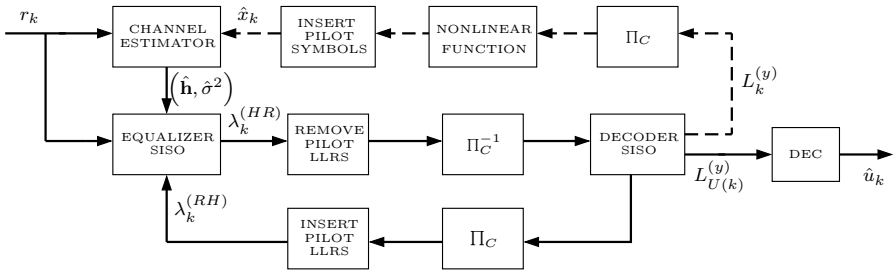

Fig. 2. Receiver architecture

A branch e for the MFSM X is defined as an (input, current state) pair (u_k, s_{k-1}) which is consistent with the behaviour of X. With each branch e we can then associate, by the defining rule (2), a unique start state $s^S(e)$, a unique end state $s^E(e)$, a unique input $u(e)$ and a unique output $x(e)$. The set of all branches for the MFSM X is denoted E_X. The indicator function for an MFSM X is denoted by

$$\text{Ind}(\mathbf{u}, \mathbf{x}, \mathbf{s} \text{ valid for } X) = T^{(X)}(\mathbf{u}, \mathbf{x}, \mathbf{s})$$

$$= \prod_{k=1}^{N} T_k^{(X)}(u_k, x_k, s_{k-1}, s_k)$$

where $T_k^{(X)}(u_k, x_k, s_{k-1}, s_k) = 1$ if and only if there exists $e \in E_X$ such that $u(e) = u_k$, $x(e) = x_k$, $s^S(e) = s_{k-1}$, $s^E(e) = s_k$.

3.2 Factor Graph

Note that the rate-1 RC encoders are MFSMs, which we denote by $R1$ and $R2$. Also, the combination of BPSK modulator and channel filter may be regarded as a "channel MFSM" H, with inputs $\{y_k\}$. Therefore, since any branch $e \in E_H$ defines a particular input and start state, we may without ambiguity define $x_j(e)$ to be the variable lying on channel tap $j \in \{0, 1, \ldots L-1\}$ for this branch. Thus we can write the channel MFSM output for branch $e \in E_H$ as

$$x(e) = \sum_{j=0}^{L-1} h_k^{(j)} x_j(e) \tag{3}$$

The state sequence followed by the channel is denoted by $\mathbf{s}''$, and the channel is assumed to start in the zero state, *i.e.* $s_0'' = 0$.

We initially approach the problem as if the channel gains $\left\{h_k^{(j)}\right\}$ and the noise variance σ^2 were known exactly (this assumption will approximately hold if the channel estimator works correctly). For MAP decoding, we wish to maximize $P(u_k|\mathbf{r})$ for each $k = 1, 2, \ldots N$. This function is the summary for u_k of

$$P(\mathbf{u}, \mathbf{a}, \mathbf{b}, \mathbf{q}, \mathbf{z}, \mathbf{s}, \mathbf{s}', \mathbf{s}''|\mathbf{r}) =$$

$$\frac{p(\mathbf{r}|\mathbf{u}, \mathbf{a}, \mathbf{b}, \mathbf{q}, \mathbf{z}, \mathbf{s}, \mathbf{s}', \mathbf{s}'')\, P(\mathbf{u}, \mathbf{a}, \mathbf{b}, \mathbf{q}, \mathbf{z}, \mathbf{s}, \mathbf{s}', \mathbf{s}'')}{p(\mathbf{r})}$$

Recall that z_k denotes the channel MFSM output as defined in (1). We define, for each $k = 1, 2, \ldots N_t$,

$$P_k(z_k) = p(r_k|z_k) = C \exp(-|z_k - r_k|^2 / (2\sigma^2)) \tag{4}$$

We assume that the probability mass function $P(\mathbf{u}, \mathbf{a}, \mathbf{b}, \mathbf{q}, \mathbf{z}, \mathbf{s}, \mathbf{s}', \mathbf{s}'')$ is uniform over all valid configurations of the variables $\mathbf{u}, \mathbf{a}, \mathbf{b}, \mathbf{q}, \mathbf{z}, \mathbf{s}, \mathbf{s}', \mathbf{s}''$ which satisfy $s_0 = 0$, $s_N = 0$, $s_0' = 0$ and $s_0'' = 0$. Then this probability mass function is proportional to

$$g(\mathbf{u}, \mathbf{a}, \mathbf{b}, \mathbf{q}, \mathbf{z}, \mathbf{s}, \mathbf{s}', \mathbf{s}'') =$$

$$\prod_{k=1}^{N_t} P_k(z_k) \prod_{k=1}^{N_q} J_k(q_k)\, T^{(R1)}(\mathbf{u}, \mathbf{a}, \mathbf{s})\, \nu_0(s_0)\, \nu_0(s_N)$$

$$\cdot T^{(R2)}(\mathbf{u}', \mathbf{b}, \mathbf{s}')\, \nu_0(s_0')\, T^{(H)}(\mathbf{y}, \mathbf{z}, \mathbf{s}'')\, \nu_0(s_0'')$$

$$= \prod_{k=1}^{N_t} P_k(z_k) \prod_{k=1}^{N_q} J_k(q_k) \prod_{k=1}^{N} T_k^{(R1)}(u_k, a_k, s_{k-1}, s_k)$$

$$\cdot \nu_0(s_0)\, \nu_0(s_N) \prod_{k=1}^{N} T_k^{(R2)}\left(u_{\Pi(k)}, b_k, s_{k-1}', s_k'\right)$$

$$\cdot \nu_0(s_0') \prod_{k=1}^{N} T_k^{(H)}\left(y_k, z_k, s_{k-1}'', s_k''\right) \nu_0(s_0'')$$

where $\nu_p(x)$ denotes the indicator function $\text{Ind}(x = p)$. The factor graph is illustrated in figure 3 for the case $N = 4$, $S_a = \{1, 3\}$, $S_b = \{2, 4\}$ and $N_q = 3$. Note that, for the purposes of factor graph theory, the variables $\mathbf{y}$ are considered to be simply a relabelling of the variables $\mathbf{u}, \mathbf{a}, \mathbf{b}, \mathbf{q}$.

3.3 Message Passing Schedule and Notation

The following outlines exactly our message passing schedule on the factor graph, and also introduces some notation. Assume at the outset that all edges have unit messages pending. Also, automatic relaying of messages by variable nodes of degree two is not mentioned explicitly, but is to be assumed throughout. The message passing schedule is as follows:

(1) Messages are passed from the following pendant factor nodes, simultaneously:

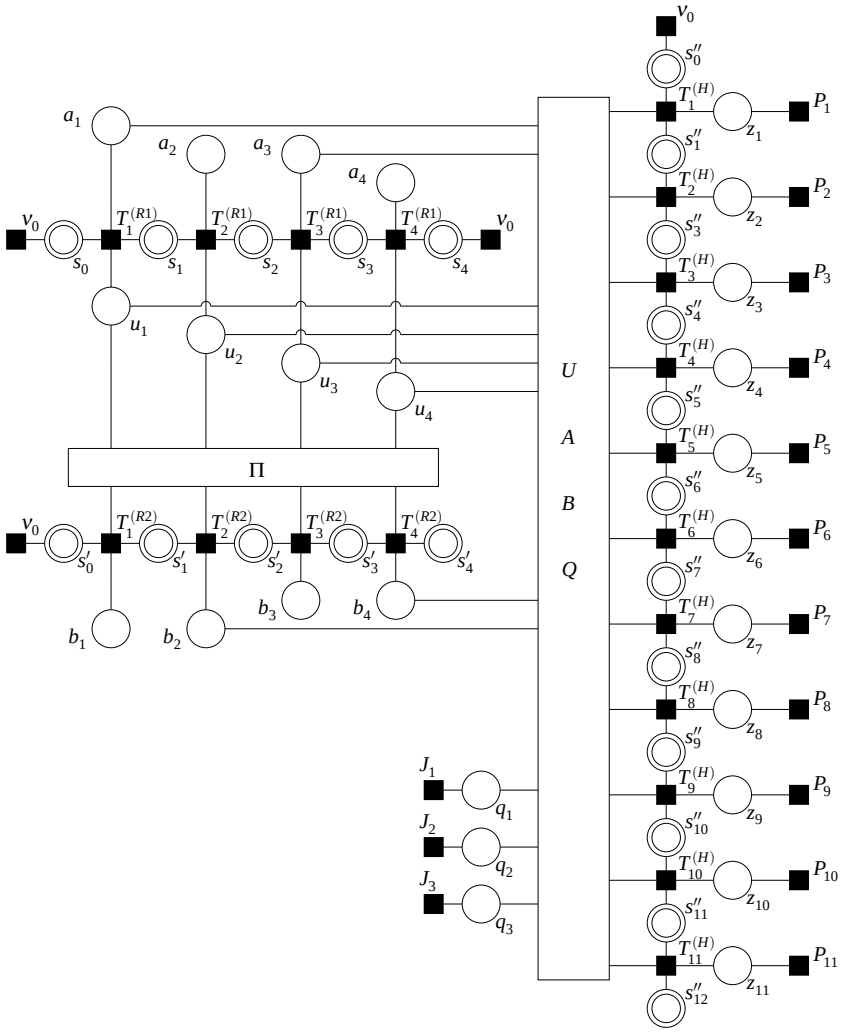

Fig. 3. Factor graph for the iterative receiver. The factor graph is illustrated for the case $N = 4$, $S_a = \{1, 3\}$, $S_b = \{2, 4\}$ and insertion of 3 pilot bits.

- $J_k \rightarrow q_k$ for each $k = 1, 2, \ldots N_q$. LLR for message = $\lambda_{Q(k)}^{(RH)}$.
- $\nu_0 \rightarrow s_0$. Log domain message = α_0.
- $\nu_0 \rightarrow s'_0$. Log domain message = α'_0.
- $\nu_0 \rightarrow s''_0$. Log domain message = α''_0.
- $\nu_0 \rightarrow s_N$. Log domain message = β_N.

For $iter = 1$ to N_{iter}

Channel section:

(2a) $P_k \rightarrow z_k$ for each $k = 1, 2, \ldots N_t$ (simultaneously).

(2b) $T_k^{(H)} \rightarrow s''_k$ for each $k = 1, 2, \ldots N_t - 1$ (sequentially). Log domain message = α''_k.

(2c) $T_k^{(H)} \rightarrow s''_{k-1}$ for each $k = N_t, N_t - 1, \ldots 2$ (sequentially). Log domain message = β''_{k-1}.

(2d) $T_{U(k)}^{(H)} \rightarrow u_k$ for each $k = 1, 2, \ldots N$ (simultaneously). LLR for message = $\lambda_{U(k)}^{(HR)}$.

(2e) $T_{A(k)}^{(H)} \rightarrow a_k$ for each $k \in S_a$ (simultaneously). LLR for message = $\lambda_{A(k)}^{(HR)}$.

(2f) $T_{B(k)}^{(H)} \rightarrow b_k$ for each $k \in S_b$ (simultaneously). LLR for message = $\lambda_{B(k)}^{(HR)}$.

PCC upper section:

(3a) $u_k \rightarrow T_k^{(R1)}$ for each $k = 1, 2, \ldots N$ (simultaneously). LLR for message = $\lambda_k^{(21)}$.

(3b) $T_k^{(R1)} \rightarrow s_k$ for each $k = 1, 2, \ldots N - 1$ (sequentially). Log domain message = α_k.

(3c) $T_k^{(R1)} \rightarrow s_{k-1}$ for each $k = N, N - 1, \ldots 2$ (sequentially). Log domain message = β_{k-1}.

(3d) $T_k^{(R1)} \rightarrow u_k$ for each $k = 1, 2, \ldots N$ (simultaneously). LLR for message = $\lambda_k^{(12e)}$.

(3e) $T_k^{(R1)} \rightarrow a_k$ for each $k \in S_a$ (simultaneously). LLR for message = $\lambda_{A(k)}^{(RH)}$.

(3f) Summarize for a_k for each $k \in S_a$ (simultaneously). LLR for summary = $L_{A(k)}^{(y)}$.

PCC lower section:

(4a) $u_k \rightarrow T^{(R2)}_{\Pi inv(k)}$ for each $k = 1, 2, \ldots N$ (simultaneously). LLR for message = $\lambda^{(12)}_k$.

(4b) $T^{(R2)}_k \rightarrow s'_k$ for each $k = 1, 2, \ldots N-1$ (sequentially). Log domain message = α'_k.

(4c) $T^{(R2)}_k \rightarrow s'_{k-1}$ for each $k = N, N-1, \ldots 2$ (sequentially). Log domain message = β'_{k-1}.

(4d) $T^{(R2)}_{\Pi inv(k)} \rightarrow u_k$ for each $k = 1, 2, \ldots N$ (simultaneously). LLR for message = $\lambda^{(21e)}_k$.

(4e) $T^{(R2)}_k \rightarrow b_k$ for each $k \in S_b$ (simultaneously). LLR for message = $\lambda^{(RH)}_{B(k)}$.

(4f) Summarize for b_k for each $k \in S_b$ (simultaneously). LLR for summary = $L^{(y)}_{B(k)}$.

Final message passing and summaries:

(5a) $u_k \rightarrow T^{(H)}_{U(k)}$ for each $k = 1, 2, \ldots N$ (simultaneously). LLR for message is = $\lambda^{(RH)}_{U(k)}$.

(5b) Summarize for u_k for each $k = 1, 2, \ldots N$ (simultaneously). LLR for summary = $L^{(y)}_{U(k)}$.

Endfor

The channel is estimated initially prior to the execution of (1), and is re-estimated after every execution of (5b). For messages which require channel knowledge, we substitute estimates. Thus we use channel estimates $\left\{\hat{h}^{(j)}_k\right\}$ in place of the actual taps in (3), and the noise variance estimate $\hat{\sigma^2}$ in place of the actual value in (4). Hence the receiver algorithm may be deduced as follows ($\max^*$ denoting the Jacobian logarithm):

INITIALIZATION:

- Perform initial channel estimation based on the receive pilot values $\left\{r_{Q(k)} | k \in S_q\right\}$ to obtain channel gain estimates $\left\{\hat{h}^{(j)}_k\right\}$ and an estimate of the AWGN noise variance, $\hat{\sigma^2}$.
- Initialize, for each $k \in S_q$

$$\lambda^{(RH)}_{Q(k)} = \begin{cases} \infty & \text{if } D(k) = 1 \\ -\infty & \text{if } D(k) = 0 \end{cases}$$

- Initialize, for each $k \in \{1, 2, \ldots N_t\} \setminus S'_q$

$$\lambda^{(RH)}_k = 0$$

- Initialize

$$\alpha_0(s_0) = \begin{cases} 0 & \text{if } s_0 = 0 \\ -\infty & \text{if } s_0 \neq 0 \end{cases}$$

$$\alpha'_0(s'_0) = \begin{cases} 0 & \text{if } s'_0 = 0 \\ -\infty & \text{if } s'_0 \neq 0 \end{cases}$$

$$\alpha''_0(s''_0) = \begin{cases} 0 & \text{if } s''_0 = 0 \\ -\infty & \text{if } s''_0 \neq 0 \end{cases}$$

$$\beta_N(s_N) = \begin{cases} 0 & \text{if } s_N = 0 \\ -\infty & \text{if } s_N \neq 0 \end{cases}$$

$$\beta'_N(s'_N) = 0 \qquad \forall s'_N$$

$$\beta''_N(s''_N) = 0 \qquad \forall s''_N$$

- Initialize, for each $k = 1, 2, \ldots N$

$$\lambda^{(21e)}_k = 0$$

For $iter = 1$ to N_{iter}

EQUALIZER SISO:

- Forward recursion: for each $k = 1, 2, \ldots N_t - 1$, and for each $s''_k \in S_H$

$$\alpha''_k(s''_k) = \max_{e|s^E(e)=s''_k}{}^* \left\{\alpha''_{k-1}\left(s^S(e)\right) - \frac{|x(e) - r_k|^2}{2\hat{\sigma^2}} + \frac{\mathcal{M}(u(e))\lambda^{(RH)}_k}{2}\right\}$$

- Backward recursion: for each $k = N_t, N_t - 1, \ldots 2$, and for each $s''_{k-1} \in S_H$

$$\beta''_{k-1}\left(s''_{k-1}\right) = \max_{e|s^S(e)=s''_{k-1}}{}^* \left\{\beta''_k\left(s^E(e)\right) - \frac{|x(e) - r_k|^2}{2\hat{\sigma^2}} + \frac{\mathcal{M}(u(e))\lambda^{(RH)}_k}{2}\right\}$$

- Compute, for each $k \in \{1, 2, \ldots N_t\} \setminus S'_q$

$$\lambda^{(HR)}_k = \max_{e|u(e)=1}{}^* \left\{\alpha''_{k-1}\left(s^S(e)\right) - \frac{|x(e) - r_k|^2}{2\hat{\sigma^2}} + \beta''_k\left(s^E(e)\right)\right\} - \max_{e|u(e)=0}{}^* \left\{\alpha''_{k-1}\left(s^S(e)\right) - \frac{|x(e) - r_k|^2}{2\hat{\sigma^2}} + \beta''_k\left(s^E(e)\right)\right\}$$

INTERLEAVER:

- Set, for each $k = 1, 2, \ldots N$,

$$\lambda^{(a)}_k = \begin{cases} \lambda^{(HR)}_{A(k)} & \text{if } k \in S_a \\ 0 & \text{if } k \notin S_a \end{cases}$$

$$\lambda^{(b)}_k = \begin{cases} \lambda^{(HR)}_{B(k)} & \text{if } k \in S_b \\ 0 & \text{if } k \notin S_b \end{cases}$$

$R1$ SISO:

- Compute, for each $k = 1, 2, \ldots N$,

$$\lambda^{(21)}_k = \lambda^{(21e)}_k + \lambda^{(HR)}_{U(k)}$$

- Forward recursion: for each $k = 1, 2, \ldots N - 1$, and for each $s_k \in S_{R1}$

$$\alpha_k(s_k) = \max_{e|s^E(e)=s_k}{}^* \left\{\alpha_{k-1}\left(s^S(e)\right) + \frac{\mathcal{M}(x(e))\lambda^{(a)}_k}{2} + \frac{\mathcal{M}(u(e))\lambda^{(21)}_k}{2}\right\}$$

- Backward recursion: for each $k = N, N-1, \ldots 2$, and for each $s_{k-1} \in S_{R1}$

$$\beta_{k-1}(s_{k-1}) = \max_{e|s^S(e)=s_{k-1}}{}^* \left\{\beta_k\left(s^E(e)\right) + \frac{\mathcal{M}(x(e))\lambda^{(a)}_k}{2} + \frac{\mathcal{M}(u(e))\lambda^{(21)}_k}{2}\right\}$$

- Extrinsic information computation: for each $k = 1, 2, \ldots N$

$$\lambda_k^{(12e)} = \max_{e|u(e)=1}{}^* \left\{ \alpha_{k-1}\left(s^S(e)\right) + \frac{\mathcal{M}(x(e))\,\lambda_k^{(a)}}{2} + \beta_k\left(s^E(e)\right) \right\} - \max_{e|u(e)=0}{}^* \left\{ \alpha_{k-1}\left(s^S(e)\right) + \frac{\mathcal{M}(x(e))\,\lambda_k^{(a)}}{2} + \beta_k\left(s^E(e)\right) \right\}$$

- Parity extrinsic information computation: for each $k \in S_a$

$$\lambda_{A(k)}^{(RH)} = \max_{e|x(e)=1}{}^* \left\{ \alpha_{k-1}\left(s^S(e)\right) + \frac{\mathcal{M}(u(e))\,\lambda_k^{(21)}}{2} + \beta_k\left(s^E(e)\right) \right\} - \max_{e|x(e)=0}{}^* \left\{ \alpha_{k-1}\left(s^S(e)\right) + \frac{\mathcal{M}(u(e))\,\lambda_k^{(21)}}{2} + \beta_k\left(s^E(e)\right) \right\}$$

- Compute, for each $k \in S_a$,

$$L_{A(k)}^{(y)} = \lambda_k^{(a)} + \lambda_{A(k)}^{(HR)}$$

$R2$ SISO:

- Compute, for each $k = 1, 2, \ldots N$

$$\lambda_k^{(12)} = \lambda_k^{(12e)} + \lambda_{U(k)}^{(HR)}$$

- Forward recursion: for each $k = 1, 2, \ldots N-1$, and for each $s'_k \in S_{R2}$

$$\alpha'_k(s'_k) = \max_{e|s^E(e)=s'_k}{}^* \left\{ \alpha'_{k-1}\left(s^S(e)\right) + \frac{\mathcal{M}(x(e))\,\lambda_k^{(b)}}{2} + \frac{\mathcal{M}(u(e))\,\lambda_{\Pi(k)}^{(12)}}{2} \right\}$$

- Backward recursion: for each $k = N, N-1, \ldots 2$, and for each $s'_{k-1} \in S_{R2}$

$$\beta'_{k-1}\left(s'_{k-1}\right) = \max_{e|s^S(e)=s'_{k-1}}{}^* \left\{ \beta'_k\left(s^E(e)\right) + \frac{\mathcal{M}(x(e))\,\lambda_k^{(b)}}{2} + \frac{\mathcal{M}(u(e))\,\lambda_{\Pi(k)}^{(12)}}{2} \right\}$$

- Extrinsic information computation: for each $k = 1, 2, \ldots N$

$$\lambda_{\Pi(k)}^{(21e)} = \max_{e|u(e)=1}{}^* \left\{ \alpha'_{k-1}\left(s^S(e)\right) + \frac{\mathcal{M}(x(e))\,\lambda_k^{(b)}}{2} + \beta'_k\left(s^E(e)\right) \right\} - \max_{e|u(e)=0}{}^* \left\{ \alpha'_{k-1}\left(s^S(e)\right) + \frac{\mathcal{M}(x(e))\,\lambda_k^{(b)}}{2} + \beta'_k\left(s^E(e)\right) \right\}$$

- Parity extrinsic information computation: for each $k \in S_b$

$$\lambda_{B(k)}^{(RH)} = \max_{e|x(e)=1}{}^* \left\{ \alpha_{k-1}\left(s^S(e)\right) + \frac{\mathcal{M}(u(e))\,\lambda_{\Pi(k)}^{(12)}}{2} + \beta_k\left(s^E(e)\right) \right\} - \max_{e|x(e)=0}{}^* \left\{ \alpha_{k-1}\left(s^S(e)\right) + \frac{\mathcal{M}(u(e))\,\lambda_{\Pi(k)}^{(12)}}{2} + \beta_k\left(s^E(e)\right) \right\}$$

- Compute, for each $k \in S_b$,

$$L_{B(k)}^{(y)} = \lambda_k^{(b)} + \lambda_{B(k)}^{(HR)}$$

FINAL COMPUTATIONS:

- Compute, for each $k = 1, 2, \ldots N$,

$$\lambda_{U(k)}^{(RH)} = \lambda_k^{(12e)} + \lambda_k^{(21e)}$$

- Compute, for each $k = 1, 2, \ldots N$,

$$L_{U(k)}^{(y)} = \lambda_k^{(12e)} + \lambda_k^{(21e)} + \lambda_{U(k)}^{(HR)}$$

- Re-estimate the channel based on $\{r_k\}$ and $\left\{L_k^{(y)}\right\}$ to obtain new estimates for the channel taps $\left\{\hat{h}_k^{(j)}\right\}$ and AWGN noise variance $\hat{\sigma^2}$.

Endfor

TERMINATION:

Finally, make decisions via

$$\hat{u}_k = \begin{cases} 1 & \text{if } L_{U(k)}^{(y)} > 0 \\ 0 & \text{if } L_{U(k)}^{(y)} \le 0 \end{cases}$$

It may easily be seen that the algorithm we have derived matches exactly that outlined in [1]. Note that the factor graph contains variable nodes corresponding to the pilot symbols. By contrast, the factor graph corresponding to the iterative receiver of [2] is the same as that for the PCC decoder alone. Also, the insertion and removal of pilot symbol LLRs has been formally justified.

4 Conclusion

This paper has presented a derivation, using factor graph theory, of an iterative receiver for PCC coded transmission over frequency selective channels. The derivation illustrates from a factor graph perspective the interaction between the channel estimation algorithm and the sum-product algorithm. The derivation explains how the need for removal and insertion of "pilot LLRs" arises naturally from the sum-product algorithm. Also, the pilot symbols were shown to appear as variable nodes in the factor graph.

References

[1] M. F. Flanagan, A. D. Fagan and M. Herro. *Pilot symbol assisted turbo equalization over a frequency selective Rayleigh fading channel*. Proc. 3rd International Symposium on Turbo Codes and Related Topics. pp. 359–362, 1-5 Sep. 2003.

[2] M. C. Valenti and B. D. Woerner. *Iterative Channel Estimation and Decoding of Pilot Symbol Assisted Turbo Codes Over Flat-Fading Channels*. IEEE Journal on Selected Areas in Communications. Vol. 19, No. 9, pp. 1697–1705, Sep. 2001.

[3] F. R. Kschischang, B. J. Frey and H. -A. Loeliger. *Factor graphs and the sum-product algorithm*. IEEE Trans. Information Theory. Vol. 47, No. 2, pp. 498–519, Feb. 2001.

[4] A. P. Worthen and W. E. Stark. *Unified Design of Iterative Receivers Using Factor Graphs*. IEEE Trans. Information Theory. Vol. 47, No. 2, pp. 843–849, Feb. 2001.

[5] C. Douillard, M. Jézéquel, C. Berrou, A. Picart, P. Didier and A. Glavieux. *Iterative Correction of Intersymbol Interference: Turbo-Equalization*. European Transactions on Telecommunications. Vol. 6, pp. 507–511, Sep.-Oct. 1995.

[6] G. D. Forney. *Codes on Graphs: Normal Realizations*. IEEE Trans. Information Theory. Vol. 47, No. 2, pp. 520–548, Feb. 2001.