

OFDM SYMBOL DETECTION WITH INTERSPERSED PILOT SYMBOLS AND CHANNEL DISTRIBUTION INFORMATION AT THE RECEIVER

Cheran M. Vithanage and Justin P. Coon

Telecommunications Research Laboratory, Toshiba Research Europe Limited,
32, Queen Square, Bristol BS1 4ND, England. Email: cheran@toshiba-trel.com

ABSTRACT

Detection of OFDM transmissions with interspersed pilot symbols is considered. A hard output detection algorithm developed by Taricco *et. al.* for flat fading channels is extended for this frequency selective scenario. In the investigated systems with 128 subcarriers, this algorithm outperforms conventional approaches and performs close to a genie aided receiver even with the use of a single pilot symbol for the whole frame. Furthermore, a novel soft output generating algorithm is developed, which is more suitable for channel coded systems. The resultant algorithm is capable of detecting coded OFDM transmissions with fewer pilot symbols than that required for noiseless channel identification.

Index Terms— Dispersive channels, Signal detection

1. INTRODUCTION

Orthogonal frequency division multiplexing (OFDM) is the system of choice for many upcoming communications standards. For coherent detection of OFDM transmissions, pilot symbols are transmitted on some of the subcarriers [1]. While a pilot-based channel estimation stage followed by a symbol detection stage based on the estimated channels is the conventional approach at the receiver [2], consideration of these two stages together leads to improved detection performance [3]. Motivated by the works of Taricco *et. al.* [4], [5] we here develop hard and soft output algorithms for such receivers. While the complexities of these algorithms are linear in the number of subcarriers, they outperform conventional receivers when the number of pilot carrying subcarriers is low.

2. PROBLEM STATEMENT

Let us consider an N -subcarrier OFDM system and let the frequency domain symbol vector for some particular OFDM frame be $\mathbf{x} = (x_1, x_2, \dots, x_N)^T$. Notations $(\cdot)^T$, $(\cdot)^H$ and $(\cdot)^{-1}$ are used to denote the transpose, conjugate transpose and the inverse of a matrix, respectively. Bold faced lower and upper case letters denote vectors and matrices, respectively. Above symbol vector is assumed to consist of both data and pilots. That is, the pilots are interspersed with the data and for convenience we assume that the first subcarrier transmits a pilot symbol. Let the data and pilot carrying subcarriers be denoted by the sets \mathcal{D} and \mathcal{P} , respectively. Then, $|\mathcal{D}| + |\mathcal{P}| = N$, where $|\cdot|$ denotes the cardinality of a set. We assume that all the subcarriers transmit symbols from some modulation alphabet \mathcal{Q} and

that the pilot subcarriers carry the symbol $\omega_{pilot} \in \mathcal{Q}$, for convenience. Also let \mathbf{x}_p denote a vector containing the symbols x_v for $v \in \mathcal{P}$.

Now let us consider the modelling of the wireless channel. We consider the frequency domain channel fading coefficient on subcarrier v to be f_v and $\mathbf{f} = (f_1, f_2, \dots, f_N)^T$. Taking the time domain channel impulse response to contain L ($< N$) symbol spaced coefficients, denoted as: $\mathbf{h}_L = (h_1, h_2, \dots, h_L)^T$, let Θ be the $N \times L$ discrete Fourier transform matrix with the (n, l) th element being $e^{-j\frac{2\pi(n-1)(l-1)}{N}}$ such that $\mathbf{f} = \Theta\mathbf{h}_L$ [3]. Let $\mathbf{y} = (y_1, y_2, \dots, y_N)^T$ and $\mathbf{n} = (n_1, n_2, \dots, n_N)^T$ be the frequency domain signal and noise vectors manifesting at the receiver, respectively. With a proper cyclic prefix at the transmitter to generate a circulant equivalent channel at the receiver, and letting $\mathbf{X} = \text{diag}(\mathbf{x})$, we have

$$\mathbf{y} = \mathbf{X}\Theta\mathbf{h}_L + \mathbf{n}. \quad (1)$$

Assuming a Rayleigh fading scenario, we model $\mathbf{h}_L = \mathbf{R}^{\frac{1}{2}}\mathbf{h}_0$ with \mathbf{h}_0 having a zero mean circularly symmetric complex Gaussian (ZM-CSCG) distribution with covariance matrix \mathbf{I}_L , where \mathbf{I}_L is the $L \times L$ identity matrix and $\mathbf{R} = E\{\mathbf{h}_L\mathbf{h}_L^H\}$. Essentially, \mathbf{R} defines the inter-tap correlations of the impulse response. For power normalisation, we take $\text{Trace}(\mathbf{R}) = 1$ and $E\{\mathbf{nn}^H\} = N_0\mathbf{I}_N$. The resultant system model is $\mathbf{y} = \mathbf{X}\Theta\mathbf{R}^{\frac{1}{2}}\mathbf{h}_0 + \mathbf{n}$. Assuming that the receiver knows the probability distribution of the channel, which is essentially determined by \mathbf{R} and N_0 , and the pilot symbols \mathbf{x}_p , we address two problems in this paper. The first is the maximum likelihood sequence detection, or the computation of

$$\arg \max_{\mathbf{x}} p(\mathbf{y} | \mathbf{x}; \mathbf{x}_p, \mathbf{R}, N_0). \quad (2)$$

The other is the computation of a *a posteriori* symbol probability distributions:

$$p(x_v | \mathbf{y}; \mathbf{x}_p, \mathbf{R}, N_0) \quad \text{for each } v \in \mathcal{D}. \quad (3)$$

In the following, we develop approximate solutions to the above problems that have linear complexity in N .

3. THE ALGORITHM FOR SEQUENCE DETECTION

Optimal sequence detection is the computation of

$$\hat{\mathbf{x}} = \arg \max_{\mathbf{x}} \mathbb{E}_{\mathbf{h}_0} p(\mathbf{y} | \mathbf{x}, \mathbf{h}_0; \mathbf{x}_p, \mathbf{R}, N_0).$$

This computation, due to its averaging over the channel distribution, indicates the avoidance of an explicit channel estimation stage. Since the additive noise is ZM-CSCG distributed,

$$\hat{\mathbf{x}} = \arg \max_{\mathbf{x}} \mathbb{E}_{\mathbf{h}_0} \left\{ \exp \left[-\frac{1}{N_0} (\mathbf{y} - \mathbf{k})^H (\mathbf{y} - \mathbf{k}) \right] \right\},$$

The authors thank the Directors of Toshiba Telecommunications Research Laboratory for their excellent support.

where $\mathbf{k} = \mathbf{X}\mathbf{O}\mathbf{R}^{\frac{1}{2}}\mathbf{h}_0$. Letting $\mathbf{b} = -\frac{\mathbf{R}^{\frac{1}{2}}\mathbf{O}^H\mathbf{X}^H\mathbf{y}}{N_0}$ and $\mathbf{A} = \frac{\mathbf{R}^{\frac{1}{2}}\mathbf{O}^H\mathbf{X}^H\mathbf{X}\mathbf{O}\mathbf{R}^{\frac{1}{2}}}{N_0}$, we have,

$$\hat{\mathbf{x}} = \arg \max_{\mathbf{x}} \mathbb{E}_{\mathbf{h}_0} \left\{ \exp \left[- \left(\mathbf{h}_0^H \mathbf{A} \mathbf{h}_0 + \mathbf{h}_0^H \mathbf{b} + \mathbf{b}^H \mathbf{h}_0 \right) \right] \right\}.$$

As shown in [4], [5], this expectation can be solved in closed form:

$$\begin{aligned} \hat{\mathbf{x}} &= \arg \max_{\mathbf{x}} \frac{\exp \{ \mathbf{b}^H (\mathbf{I}_L + \mathbf{A})^{-1} \mathbf{b} \}}{\det (\mathbf{I}_L + \mathbf{A})} \\ &= \arg \max_{\mathbf{x}} \left\{ \mathbf{b}^H (\mathbf{I}_L + \mathbf{A})^{-1} \mathbf{b} - \log \det (\mathbf{I}_L + \mathbf{A}) \right\}. \end{aligned} \quad (4)$$

As illustrated in [3] for constant amplitude modulations, exact computation of (4) incurs a complexity which is exponential in N . In [6], this complexity was avoided by resorting to a sub-optimal per symbol decoding scheme. Here we follow the approach of [4], to develop a sequential algorithm to find this maximum likelihood solution. With a Markovian assumption, this approach enabled the development of a $\mathcal{O}(N)$ complexity algorithm with excellent performance. We assume that $\mathbf{R}^{\frac{1}{2}}$ is invertible, although this condition can be relaxed since the final algorithms do not involve any explicit matrix inversions. We use notation $\mathbf{x}_{1:v}$ to denote the vector $(x_1, x_2, \dots, x_v)^T$. With $\mathbf{K} = \frac{\mathbf{O}^H\mathbf{X}^H\mathbf{X}\mathbf{O}}{N_0}$ and

$$\begin{aligned} \zeta(\mathbf{x}_{1:N}) &= \frac{1}{N_0^2} \mathbf{y}^H \mathbf{X} \mathbf{O} (\mathbf{R}^{-1} + \mathbf{K})^{-1} \mathbf{O}^H \mathbf{X}^H \mathbf{y} \\ \lambda(\mathbf{x}_{1:N}) &= (-1) \log \det (\mathbf{R}^{-1} + \mathbf{K}), \end{aligned}$$

the objective function to be maximised is $\mu(\mathbf{x}_{1:N}) = \zeta(\mathbf{x}_{1:N}) + \lambda(\mathbf{x}_{1:N})$. For the development of a sequential maximiser for this objective function, for each $v \in \{1, \dots, N\}$, we need to find a breakdown such that

$$\mu(\mathbf{x}_{1:v}) = \mu(\mathbf{x}_{1:v-1}) + \Delta\mu(\mathbf{x}_{1:v-1}, x_v). \quad (5)$$

The next Section builds a set of recursions which facilitates the sequential computations of the above metric differences $\Delta\mu(\mathbf{x}_{1:v-1}, x_v)$ for each $v \in \{1, \dots, N\}$.

3.1. Set of recursions

Decomposing the matrix \mathbf{O} as $\mathbf{O}^H = [\boldsymbol{\theta}_1 \boldsymbol{\theta}_2 \dots \boldsymbol{\theta}_N]$, let $\boldsymbol{\Theta}_v$ be defined by $\boldsymbol{\Theta}_v^H = [\boldsymbol{\theta}_1 \dots \boldsymbol{\theta}_v]$. With $\mathbf{X}_v = \text{diag}(\mathbf{x}_{1:v})$ and $\mathbf{K}_v = \frac{\boldsymbol{\Theta}_v^H \mathbf{X}_v^H \mathbf{X}_v \boldsymbol{\Theta}_v}{N_0}$, we have

$$\mu(\mathbf{x}_{1:v}) = \zeta(\mathbf{x}_{1:v}) + \lambda(\mathbf{x}_{1:v}). \quad (6)$$

Here,

$$\begin{aligned} \zeta(\mathbf{x}_{1:v}) &= \frac{1}{N_0^2} \mathbf{y}_{1:v}^H \mathbf{X}_v \boldsymbol{\Theta}_v (\mathbf{R}^{-1} + \mathbf{K}_v)^{-1} \boldsymbol{\Theta}_v^H \mathbf{X}_v^H \mathbf{y}_{1:v} \\ \lambda(\mathbf{x}_{1:v}) &= -\log \det (\mathbf{R}^{-1} + \mathbf{K}_v). \end{aligned}$$

The log-determinant term can be rewritten as

$$\begin{aligned} \lambda(\mathbf{x}_{1:v}) &= \\ &= -\log \det \left(\left(\mathbf{R}^{-1} + \frac{1}{N_0} \sum_{i=1}^{v-1} \boldsymbol{\theta}_i \boldsymbol{\theta}_i^H |x_i|^2 \right) + \boldsymbol{\theta}_v \boldsymbol{\theta}_v^H |x_v|^2 \right). \end{aligned}$$

Using the determinant lemma [7]

$$\det (\mathbf{A} + \mathbf{u}\mathbf{v}^H) = \left(1 + \mathbf{v}^H \mathbf{A}^{-1} \mathbf{u} \right) \det (\mathbf{A}),$$

and letting $\boldsymbol{\Omega}(\mathbf{x}_{1:v-1}) = \left(\mathbf{R}^{-1} + \frac{1}{N_0} \sum_{i=1}^{v-1} \boldsymbol{\theta}_i \boldsymbol{\theta}_i^H |x_i|^2 \right)^{-1}$,

$$\lambda(\mathbf{x}_{1:v}) = \lambda(\mathbf{x}_{1:v-1}) - \log \left(1 + \frac{|x_v|^2}{N_0} \boldsymbol{\theta}_v^H \boldsymbol{\Omega}(\mathbf{x}_{1:v-1}) \boldsymbol{\theta}_v \right).$$

Using the matrix inversion lemma [7]

$$\left(\mathbf{A} + \mathbf{u}\mathbf{v}^H \right)^{-1} = \mathbf{A}^{-1} - \frac{\mathbf{A}^{-1} \mathbf{u}\mathbf{v}^H \mathbf{A}^{-1}}{1 + \mathbf{v}^H \mathbf{A}^{-1} \mathbf{u}},$$

and defining $\boldsymbol{\Xi}(\mathbf{x}_{1:v}) = \frac{|x_v|^2}{N_0} \frac{\boldsymbol{\Omega}(\mathbf{x}_{1:v-1}) \boldsymbol{\theta}_v \boldsymbol{\theta}_v^H \boldsymbol{\Omega}(\mathbf{x}_{1:v-1})}{1 + \frac{|x_v|^2}{N_0} \boldsymbol{\theta}_v^H \boldsymbol{\Omega}(\mathbf{x}_{1:v-1}) \boldsymbol{\theta}_v}$, we can also write,

$$\begin{aligned} \boldsymbol{\Omega}(\mathbf{x}_{1:v}) &= \left(\left(\mathbf{R}^{-1} + \frac{1}{N_0} \sum_{i=1}^{v-1} \boldsymbol{\theta}_i \boldsymbol{\theta}_i^H |x_i|^2 \right) + \boldsymbol{\theta}_v \boldsymbol{\theta}_v^H |x_v|^2 \right)^{-1} \\ &= \boldsymbol{\Omega}(\mathbf{x}_{1:v-1}) - \boldsymbol{\Xi}(\mathbf{x}_{1:v}). \end{aligned}$$

Now, consider the $\zeta(\mathbf{x}_{1:v})$ term. With $\mathbf{r}_{1:v} = \mathbf{X}_v^H \mathbf{y}_{1:v}$, we have

$$\zeta(\mathbf{x}_{1:v}) = \frac{1}{N_0^2} \mathbf{r}_{1:v}^H \boldsymbol{\Theta}_v \{ \boldsymbol{\Omega}(\mathbf{x}_{1:v-1}) - \boldsymbol{\Xi}(\mathbf{x}_{1:v}) \} \boldsymbol{\Theta}_v^H \mathbf{r}_{1:v}.$$

Letting $\mathbf{c}(\mathbf{x}_{1:v}) = \boldsymbol{\Theta}_v^H \mathbf{r}_{1:v} = \mathbf{c}(\mathbf{x}_{1:v-1}) + x_v^* y_v \boldsymbol{\theta}_v$, one can also expand the difference $N_0^2 \{ \zeta(\mathbf{x}_{1:v}) - \zeta(\mathbf{x}_{1:v-1}) \}$ as

$$\begin{aligned} |r_v|^2 \boldsymbol{\theta}_v^H \boldsymbol{\Omega}(\mathbf{x}_{1:v-1}) \boldsymbol{\theta}_v + 2\Re \left(\mathbf{c}(\mathbf{x}_{1:v-1})^H \boldsymbol{\Omega}(\mathbf{x}_{1:v-1}) \boldsymbol{\theta}_v r_v \right) \\ - \mathbf{c}(\mathbf{x}_{1:v})^H \boldsymbol{\Xi}(\mathbf{x}_{1:v}) \mathbf{c}(\mathbf{x}_{1:v}). \end{aligned}$$

Finally, using (5), (6), and the above derivations, the set of recursions:

$$\mathbf{c}(\mathbf{x}_{1:v}) = \mathbf{c}(\mathbf{x}_{1:v-1}) + x_v^* y_v \boldsymbol{\theta}_v \quad (7)$$

$$\boldsymbol{\Xi}(\mathbf{x}_{1:v}) = \frac{|x_v|^2}{N_0} \frac{\boldsymbol{\Omega}(\mathbf{x}_{1:v-1}) \boldsymbol{\theta}_v \boldsymbol{\theta}_v^H \boldsymbol{\Omega}(\mathbf{x}_{1:v-1})}{1 + \frac{|x_v|^2}{N_0} \boldsymbol{\theta}_v^H \boldsymbol{\Omega}(\mathbf{x}_{1:v-1}) \boldsymbol{\theta}_v} \quad (8)$$

$$\boldsymbol{\Omega}(\mathbf{x}_{1:v}) = \boldsymbol{\Omega}(\mathbf{x}_{1:v-1}) - \boldsymbol{\Xi}(\mathbf{x}_{1:v}), \quad (9)$$

enables the updating of the metric difference of (5) as

$$\Delta\mu(\mathbf{x}_{1:v-1}, x_v) = \frac{1}{N_0^2} \{ l + m - y \} - \log(1 + z). \quad (10)$$

Here,

$$\begin{aligned} l &= |x_v|^2 |y_v|^2 \boldsymbol{\theta}_v^H \boldsymbol{\Omega}(\mathbf{x}_{1:v-1}) \boldsymbol{\theta}_v \\ m &= 2\Re \left(\mathbf{c}(\mathbf{x}_{1:v-1})^H \boldsymbol{\Omega}(\mathbf{x}_{1:v-1}) \boldsymbol{\theta}_v x_v^* y_v \right) \\ y &= \mathbf{c}(\mathbf{x}_{1:v})^H \boldsymbol{\Xi}(\mathbf{x}_{1:v}) \mathbf{c}(\mathbf{x}_{1:v}) \\ z &= \frac{|x_v|^2}{N_0} \boldsymbol{\theta}_v^H \boldsymbol{\Omega}(\mathbf{x}_{1:v-1}) \boldsymbol{\theta}_v. \end{aligned}$$

3.2. Markovian assumption and the Viterbi algorithm based decoder

Since the domain of $\mathbf{x}_{1:v}$ grows exponentially with v , complexity of the above sequential algorithm is still exponential in N . This growing complexity can be avoided by making a type of Markovian assumption and considering that the metric to be maximised at each subcarrier depends only on the variables pertaining to that and the previous subcarriers. In other words, at $v > 1$,

the quantities $\Omega(\mathbf{x}_{1:v-1})$, $\Xi(\mathbf{x}_{1:v})$, $\Omega(\mathbf{x}_{1:v})$, $\mathbf{c}(\mathbf{x}_{1:v-1})$, $\mathbf{c}(\mathbf{x}_{1:v})$ and $\Delta\mu(\mathbf{x}_{1:v-1}, x_v)$ are substituted by $\Omega(x_{v-1})$, $\Xi(x_{v-1}, x_v)$, $\Omega(x_{v-1}, x_v)$, $\mathbf{c}(x_{v-1})$, $\mathbf{c}(x_{v-1}, x_v)$ and $\Delta\mu(x_{v-1}, x_v)$, respectively. These assumptions enable a maximiser for the objective function $\mu(\mathbf{x}_{1:N})$, to be found using the well known Viterbi algorithm.

In application, the decoding can be distributed among the *windows* across the subcarriers, determined such that the pilot symbols are always placed at the ends of a window. Assuming 1 and N_1 are the first and second pilot carrying subcarriers, the decoding algorithm is given below for the window of subcarriers indexed by $v \in \{1, 2, \dots, N_1\}$. Other windows determined by the pilot symbol placement can be similarly decoded.

For $x_1 \in \mathcal{Q}$ (Initialisation)

$$\mathbf{c}(x_1) = \omega_{pilot}^* y_1 \theta_1$$

$$\Omega(x_1) = \mathbf{R} - \frac{|\omega_{pilot}|^2}{N_0} \frac{\mathbf{R} \theta_1 \theta_1^H \mathbf{R}}{1 + \frac{|\omega_{pilot}|^2}{N_0} \theta_1^H \mathbf{R} \theta_1}$$

$$\mu(x_1) = \begin{cases} \mu_0 \gg 1 & , x_1 = \omega_{pilot} \\ 0 & , \text{otherwise} \end{cases}$$

$$\mathbf{s}(x_1) = x_1 .$$

End For

For $v = 2, 3, \dots, N_1$ (Recursion)

For $x_v \in \mathcal{Q}$

For $x_{v-1} \in \mathcal{Q}$

Compute $\mathbf{c}(x_{v-1}, x_v)$, $\Xi(x_{v-1}, x_v)$, $\Omega(x_{v-1}, x_v)$ and $\Delta\mu(x_{v-1}, x_v)$ using (7), (8), (9), and (10).

$$\mu(x_{v-1}, x_v) = \mu(x_{v-1}) + \Delta\mu(x_{v-1}, x_v)$$

End For

$$\hat{x}_{v-1} = \arg \max_{x_{v-1} \in \mathcal{Q}} \mu(x_{v-1}, x_v)$$

$$\mu(x_v) = \mu(\hat{x}_{v-1}, x_v)$$

$$\Omega(x_v) = \Omega(\hat{x}_{v-1}, x_v)$$

$$\mathbf{c}(x_v) = \mathbf{c}(\hat{x}_{v-1}, x_v)$$

$$\mathbf{s}(x_v) = (\mathbf{s}(\hat{x}_{v-1})^T x_v)^T$$

End For

End For

Output $\mathbf{s}(x_{N_1} = \omega_{pilot})$ (Decisions)

4. A SOFT-OUTPUT SYMBOL DETECTION ALGORITHM

From the Markovian assumption of Section 3.2, we have:

$$\mu(\mathbf{x}_{1:N}) \approx \mu(x_1) + \sum_{v=2}^N \Delta\mu(x_{v-1}, x_v).$$

Plugging this into the original joint probability distribution, and then using Bayes formula,

$$p(\mathbf{x} | \mathbf{y}; \mathbf{x}_p, \mathbf{R}, N_0) \approx q(\mathbf{x})$$

$$\propto \exp\{\mu(x_1)\} \prod_{v=2}^N \{\exp\{\Delta\mu(x_{v-1}, x_v)\} p(x_v)\}.$$

Essentially, the actual posterior joint distribution has been approximated by a joint distribution $q(\mathbf{x})$, in which we have $q(x_v | \mathbf{x}_{1:v-1}) = q(x_v | x_{v-1})$. Thus, there exists an underlying hidden Markov model governing $q(\mathbf{x})$. Therefore, one can use the BCJR (a.k.a. MAP) algorithm [8], to output the marginal probability distributions

of the symbols; $q(x_v)$. These outputs can be taken as the approximate posterior marginals, $p(x_v | \mathbf{y}; \mathbf{x}_p, \mathbf{R}, N_0)$, and hence the soft outputs of the algorithm.

The pseudocode of the resulting soft-output algorithm (again for the first window) is given below. The intermediate computations α , β and γ have their usual meanings in the context of the BCJR algorithm. Note also that we have used a max-log [9] type approximation in the forward recursion, to adapt the recursive computations given in Section 3.1.

For $x_1 \in \mathcal{Q}$ (Forward initialisation)

$$\mathbf{c}(x_1) = \omega_{pilot}^* y_1 \theta_1$$

$$\Omega(x_1) = \mathbf{R} - \frac{|\omega_{pilot}|^2}{N_0} \frac{\mathbf{R} \theta_1 \theta_1^H \mathbf{R}}{1 + \frac{|\omega_{pilot}|^2}{N_0} \theta_1^H \mathbf{R} \theta_1}$$

$$\alpha(x_1) = \begin{cases} 1 & , x_1 = \omega_{pilot} \\ 0 & , \text{otherwise} \end{cases}$$

End For

For $v = 2, 3, \dots, N_1$ (Forward recursion)

For $x_v \in \mathcal{Q}$

For $x_{v-1} \in \mathcal{Q}$

Compute $\mathbf{c}(x_{v-1}, x_v)$, $\Xi(x_{v-1}, x_v)$,

$\Omega(x_{v-1}, x_v)$ and $\Delta\mu(x_{v-1}, x_v)$

using (7), (8), (9), and (10).

$$\gamma(x_{v-1}, x_v) = \exp\{\Delta\mu(x_{v-1}, x_v)\} p(x_v)$$

$$\tilde{\alpha}(x_{v-1}, x_v) = \alpha(x_{v-1}) \gamma(x_{v-1}, x_v)$$

End For

$$\alpha(x_v) = \sum_{x_{v-1} \in \mathcal{Q}} \tilde{\alpha}(x_{v-1}, x_v)$$

$$\hat{x}_{v-1} = \arg \max_{x_{v-1} \in \mathcal{Q}} \tilde{\alpha}(x_{v-1}, x_v)$$

$$\Omega(x_v) = \Omega(\hat{x}_{v-1}, x_v)$$

$$\mathbf{c}(x_v) = \mathbf{c}(\hat{x}_{v-1}, x_v)$$

End For

End For

$$\beta(x_{N_1}) = \begin{cases} 1 & , x_{N_1} = \omega_{pilot} \\ 0 & , \text{otherwise} \end{cases} \quad (\text{Backward initialisation})$$

For $v = N_1 - 1, N_1 - 2, \dots, 1$ (Backward recursion)

For $x_v \in \mathcal{Q}$

$$\beta(x_v) = \sum_{x_{v+1} \in \mathcal{Q}} \beta(x_{v+1}) \gamma(x_v, x_{v+1})$$

End For

Output the distribution:

$$q(x_{v+1}) \propto \sum_{x_v \in \mathcal{Q}} \alpha(x_v) \gamma(x_v, x_{v+1}) \beta(x_{v+1})$$

End For

5. FURTHER DISCUSSION

Complexities of the above algorithms can be seen to be $\mathcal{O}(L^2 |\mathcal{Q}|^2 N)$. Thus the overall algorithm complexity is linear in N , which is much preferable to optimal algorithms for the hard or soft output generation, which have complexities that are exponential in N . While this exponential complexity has been avoided in [3] by the use of sphere decoders, complexities of such approaches are still of at least $\mathcal{O}(N^3)$ for reasonable decoding performance.

In the forward initialisation stages of the pseudocodes given earlier, no information from previous windows are considered. This enables parallel decoding across windows determined by the pilot symbol placement. An alternative is to initialise each window based on the final computations of the preceding window. In simulations, such a linked decoding scheme is shown to perform better when the number of pilot symbols is low, although such comparisons are not shown here due to the lack of space.

6. SIMULATION RESULTS

In the following simulations, an $N = 128$ subcarrier system is considered. The channel impulse responses are simulated to constitute of $L = 8$ uncorrelated taps having zero mean and equal variances. The P ($= |\mathcal{P}|$) pilot symbols were placed on the subcarrier indices $\left\{1, \frac{128}{P} + 1, \dots, \frac{(P-1)128}{P} + 1\right\}$. The modulation alphabet was QPSK. Performances of the proposed algorithms are compared against a genie aided receiver, which without transmitting any pilot symbols, is assumed to know the intermediate channels perfectly; and a “conventional algorithm” [2]. In this conventional scheme, the pilot symbols are first utilised to obtain least-squares channel estimates on those subcarriers and then these estimates are interpolated using a discrete Fourier transform based technique to obtain the channel estimates on the data carrying subcarriers. Fig. 1 illustrates

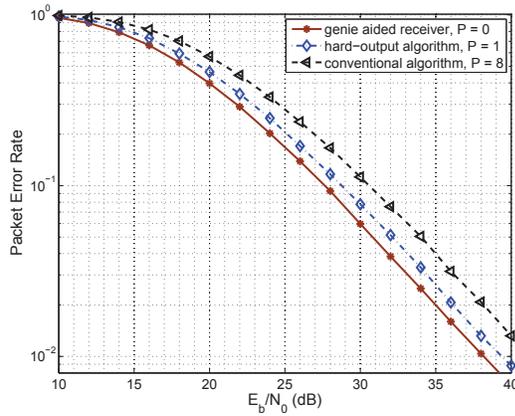


Fig. 1. Uncoded PER performance of the hard-output algorithm.

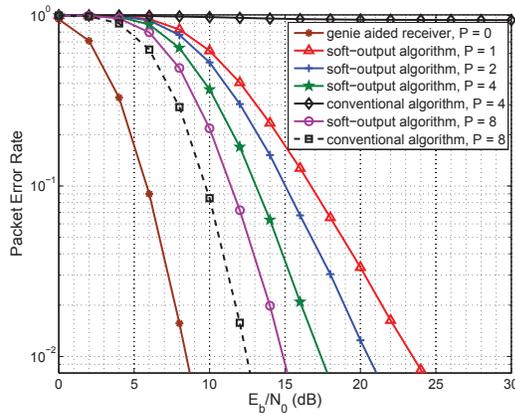


Fig. 2. Coded PER performance of the soft-output algorithm.

the performance of the hard-output sequence detection algorithm in a system without channel coding and Fig. 2 illustrates the performance of the soft-output algorithm in a system with channel coding. The channel code in this case was a rate $\frac{1}{2}$ convolutional code with constraint length 7. The packet-error-rate (PER) performance is plotted against the ratio of transmitted energy per data bit to the

noise variance at the receiver (E_b/N_0), thus accounting for the loss in efficiency due to the transmission of P pilot symbols. Each packet contained a single OFDM frame of symbols.

The sequence detection algorithm performs close to the genie aided receiver with the use of just a single pilot symbol and outperforms conventional approaches which use many more pilots. The main attraction of the soft-output algorithm is that it enables the decoding of transmissions with the number of pilots $P < L$, where conventional approaches fail to give meaningful decodings. While the decoded PER for such instances is still worse than that of the genie aided receiver, one solution to improve performance is to iterate this soft-input soft-output algorithm with the channel decoding algorithm or to utilise this decoding as the initial stage of an iterative receiver, which iterates between channel estimation, data detection and channel decoding (see, for example, [6]).

7. CONCLUSIONS

We considered the detection of OFDM symbols with interspersed pilot symbols. Assuming channel distribution information at the receiver, the problems of optimal sequence and symbol detections were considered. Motivated by the approaches of Taricco *et al.* for flat fading channels, sub-optimal algorithms were developed. These are seen to outperform conventional approaches to OFDM symbol detection, especially when the number of pilot symbols is low; despite their complexities being linear in the number of subcarriers.

8. REFERENCES

- [1] J. J. van de Beek and O. Edfors and M. Sandell and S. K. Wilson and P. O. Borjesson, “On channel estimation in OFDM systems,” in *Proc. IEEE Vehicular Technology conference*, July 1995, pp. 815–819.
- [2] H. Minn and V. K. Bhargava, “An investigation into time-domain approach for OFDM channel estimation,” *IEEE Transactions on Broadcasting*, vol. 46(4), pp. 240–248, December 2000.
- [3] T. Cui and C. Tellambura, “Joint data detection and channel estimation for OFDM systems,” *IEEE Transactions on Communications*, vol. 54(4), pp. 670–679, April 2006.
- [4] G. Taricco and E. Biglieri, “Space-time decoding with imperfect channel estimation,” *IEEE Transactions on Wireless Communications*, vol. 4(4), pp. 1874–1888, July 2005.
- [5] G. Taricco and G. Coluccia, “Optimum receiver design for correlated Rician fading MIMO channels with pilot-aided detection,” *IEEE Journal on Selected Areas in Communications*, vol. 25(7), pp. 1311–1321, September 2007.
- [6] J. Zhang, V. M. Baronkin, and Y. V. Zakharov, “Optimal detector of ofdm signals for imperfect channel estimation,” in *Proc. European signal processing conference*, Lausanne, August 2008.
- [7] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge University Press, 1985.
- [8] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, “Optimal decoding of linear codes for minimizing symbol error rate,” *IEEE Transactions on Information Theory*, vol. 20, pp. 284–287, March 1974.
- [9] P. Robertson, P. Villebrun, and P. Hoeher, “A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain,” in *Proc. IEEE International Conference on Communications*, Seattle, Washington, June 1995, pp. 1009–1013.