Andrieu, C., McGeehan, J. P., & Piechocki, R. J. (2003). Particle smoothing techniques with turbo principle for MIMO systems. In 4th IEEE Workshop on Signal Processing Advances in Wireless Communications, 2003 (SPAWC 2003). (pp. 561 - 564). Institute of Electrical and Electronics Engineers (IEEE).

Link to publication record in Explore Bristol Research
PDF-document

University of Bristol - Explore Bristol Research

General rights

This document is made available in accordance with publisher policies. Please cite only the published version using the reference above. Full terms of use are available: http://www.bristol.ac.uk/pure/about/ebr-terms.html

Take down policy

Explore Bristol Research is a digital archive and the intention is that deposited content should not be removed. However, if you believe that this version of the work breaches copyright law please contact open-access@bristol.ac.uk and include the following information in your message:

- Your contact details
- Bibliographic details for the item, including a URL
- An outline of the nature of the complaint

On receipt of your message the Open Access Team will immediately investigate your claim, make an initial judgement of the validity of the claim and, where appropriate, withdraw the item in question from public view.

**2003 4th IEEE Workshop on Signal Processing
Advances in Wireless Communications**

PARTICLE SMOOTHING TECHNIQUES WITH TURBO PRINCIPLE FOR MIMO SYSTEMS

Christophe Andrieu[1], *Robert J. Piechocki*[2,3] *and Joe P. McGeehan*[2,3]

[1]Department of Mathematics, Statistics Group, University of Bristol, BS8 1TW, UK.
[2] University of Bristol Centre For Communications Research, Woodland Road, BS8 1UB, UK.
[3] Toshiba, Telecommunications Research Laboratory, 32 Queen Square Bristol BS1 4ND, UK.
Emails: c.andrieu@bris.ac.uk - r.j.piechocki@bris.ac.uk - joe.mcgeehan@toshiba-trel.com

ABSTRACT

We address the problem of space-time communications in the context where the turbo principle is used in order to improve the robustness of the transmission. Recovering the sequence of emitted symbols in the wide-band context is known to lead to a computationally very challenging problem. We propose here an adaptive numerical approximation of the BCJR algorithm, based on particle filtering/smoothing techniques. When combined with classical estimation techniques for the channel and variance of the observation noise, this algorithm is very efficient, and scales very favourably with the dimension of the problem. This is demonstrated through numerical simulations.

1 Introduction

In this paper we present an algorithm for the recovery of symbols in the context of MIMO transmission where a coding/decoding scheme that relies on the turbo principle is used. In this context good performance is expected, but the turbo principle requires the evaluation of the so-called marginal posterior distributions, *i.e.* the marginal posterior distributions of the symbols given the observations. Direct evaluation of these quantities is not possible even in very simple cases. Indeed, even the use of the efficient and exact BCJR algorithm [2] as its complexity scales exponentially with the number of transmit antenna and the length of the transmission channel. The algorithm that we present here can be understood as being an adaptive numerical approximation of the BCJR algorithm, which relies on particle filtering techniques [3]. We use a modification of the algorithm presented in [4], which takes advantage of the finite and discrete nature of the transmitted signal and introduces the use of tempered weights. This allows for a better interaction between the turbo-decoder and the exploration algorithm, while significantly reducing the complexity of the algorithm. The procedure we propose some similarities with popular complexity reduction algorithms (*e.g.* [6], [7]), which are however designed to find maximum joint a posterior distributions of the sequence of symbols, whereas we are here interested in the marginal posterior distribution. The paper is organised as follows. In Section 2 we introduce the notation, describe the telecommunication system and state the statistical inference aims in Section 3. In Section 4 we present our algorithm and in Section 5 we demonstrate its efficiency on an example.

2 Model description and aims

2.1 The channel encoder

We consider the multiple source digital signalling problem over time dispersive channels. The binary stream of data $(b_t, t \geq 1)$ is first transformed by a channel encoder to obtain an encoded (redundant) sequence $(d_t, t \geq 1)$. A coding scheme that facilitates so-called SISO (soft-input-soft-output) decoding is used the channel encoder. Typically convolutional or LDPC (Low Density Patiry Check) coding schemes are used for that purpose. In case of LDPC the interleaveing is actually not essential since LDPC encoding implicitly involves permutation.The encoded sequence then undergoes a permutation π, leading to $(d_{\pi(t)}, t \geq 1)$ which is then mapped to a sequence $(s_t, t \geq 1)$ of digital modulation symbols. These symbols are assumed to take their values in an alphabet $\mathcal{A} = \{a_1, \ldots, a_{n_s}\} \subset \mathbb{C}^{n_s}$ that contains n_s symbols. This is summarized in Fig. 1. In order to improve the spectral efficiency the modulated sequence is divided into m parallel streams that are transmitted simultaneously from m transmit antennas.

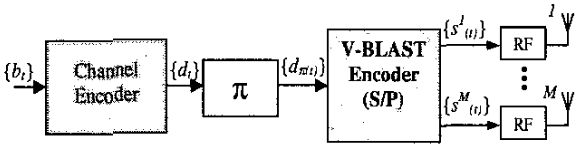

Figure 1: Schematic of the transmitter.

2.2 The equivalent channel

The signal is transmitted through a medium which introduces both delay and attenuation, and an additive noise. More precisely we model the combined effect of the pulse shaping, the transmission channel, the receive filter between the m transmitted antennas and n receive antennas with a multidimensional FIR filter. The observations are not perfect, but corrupted with an additive, temporally and spatially white Gaussian noise sequence $(\xi_t^j, t \geq 1, j = 1, \ldots, n)$. More precisely, let us introduce

$$x_t^i = (s_t^i, s_{t-1}^i, \ldots, s_{t-L+1}^i)^{\mathrm{T}},$$

0-7803-7858-X/03/$17.00 ©2003 IEEE

and

$$x_t = (x_t^1, x_t^2, \ldots, x_t^m)^{\mathrm{T}}.$$

Then the signal observed at antenna j ($j = 1, \ldots, m$) at time t is

$$y_t^j = \sum_{i=1}^{m} h_{i,j}^{\mathrm{T}} x_t^i + \xi_t^j = h_j^{\mathrm{T}} x_t + \xi_t^j = z_t^j + \xi_t^j. \tag{1}$$

where L is the length of the filters from transmit to receive antennas, assumed to be independent of i, j. The series $(\xi_t^j, t \geq 1, j = 1, \ldots, n)$ is assumed circular zero mean i.i.d. and normally distributed, *i.e.*

$$\xi_t^j \overset{i.i.d.}{\sim} \mathcal{N}_c\left(0, \sigma^2/2\right). \tag{2}$$

We choose to reformulate this problem in state-space form. First it is clear that

$$p\left(x_{t+1}|x_t\right) = p\left(s_{t+1}|s_t\right) \mathbb{I}_{[x_t]_{1:L-1}}([x_{t+1}]_{2:L}),$$

where the indicator is 0 when x_{t+1} and x_t do not have compatible "histories" and 1 otherwise. In most cases this probability distribution will simplify to

$$p\left(x_{t+1}|x_t\right) = p\left(s_{t+1}\right) \mathbb{I}_{[x_t]_{1:L-1}}([x_{t+1}]_{2:L}), \tag{3}$$

that is s_{t+1} does not depend on the past values of the symbols already transmitted. The prior $p(s_t)$ may be some prior information provided by the output of a turbo-decoder for example. In any case it should be clear that $(x_t, t \geq 1)$ is a Markov chain. Now at time $t+1$,

$$p\left(y_{t+1}|x_{t+1}\right) = \mathcal{N}_c\left(y_{t+1}; z_{t+1}, \sigma^2/2I_n\right),$$

where for any integer a, I_a is the a × a identity matrix and

$$z_{t+1} = (z_{t+1}^1, z_{t+1}^2, \ldots, z_{t+1}^n)^{\mathrm{T}}.$$

3 Estimation objectives and computational issues

Here, for the sake of simplicity, we will assume that the noise level and the transfert function are known. The conditioning of all distribution on these values is implicit everywhere. The estimation of the parameters is routine and briefly discussed at the end of the next section.

3.1 Estimation purposes

The turbo decoder requires the evaluation of the family of T posterior marginal distributions of the symbols given the observations, in other words $p(s_t|y_{1:T})$ for $t = 1, \ldots, T$. Whereas it is relatively easy to evaluation the joint posterior distribution of the channels , the variance of the observation noise and the symbols, as it is a simple by-product of the application of Bayes' rule, it is much more difficult in practice to estimate the marginal posterior distributions of the symbols. Indeed, for any $t = 1, \ldots, T$, the marginal posterior distribution is equal to

$$p\left(s_t^{(i)} \middle| y_{1:T}\right) = \sum_{\{s_u^v; u \neq t \,\&\, v \neq i\}} p\left(s_{1:T}^{(1:m)} | y_{1:T}\right).$$

To give an idea about the complexity involved with the evaluation of this quantity we consider a simple scenario. Let the modulation be a BPSK, $T = 100$, $m = n = 2$ (*i.e.* two sources and two antennas). In this simple case the number of discrete terms in the sum is $2^{2\times 99}$, *i.e.* a number of term of order 10^{30}! Therefore systematic evaluation of the discrete sum is impossible. The forward backward algorithm is a way of reducing this complexity. However, as we shall see, when the quantity n_s^{Lm} is too large it is still necessary to resort to numerical approximations.

3.2 The BCJR algorithm

The BCJR algorithm [2], or forward-backward algorithm, is an efficient way of calculating the marginal distributions $(p(x_t|y_{1:t}), 1 \leq t \leq T)$, using a so-called sum-product algorithm. We recall here the derivation of the algorithm, and point out the combinatorial difficulties inherent to this algorithm. Recall that (note that we here use the notation $\int$ but that most of the sums are in fact discreet). First we notice that

$$\begin{aligned} p(x_t|y_{1:T}) &= \sum_{x_{t+1}} p(x_t, x_{t+1}|y_{1:T}) \\ &= \sum_{x_{t+1}} p(x_t|x_{t+1}, y_{1:T}) p(x_{t+1}|y_{1:T}) \\ &= \sum_{x_{t+1}} p(x_t|x_{t+1}, y_{1:t}) p(x_{t+1}|y_{1:T}) \\ &= \sum_{x_{t+1}} \frac{p(x_t|y_{1:t}) p\left(x_{t+1}|x_t\right)}{p(x_{t+1}|y_{1:t})} p(x_{t+1}|y_{1:T}), \end{aligned}$$

where the sum is over all possible values of x_{t+1}. The expression suggests therefore the following filtering-forward/backward-smoothing algorithm (which aims at recursively computing $p(x_t|y_{1:T})$).

Forward filtering

1. *Initialization*: Compute $p\left(x_0\right)$ the distribution of the initial state for all possible values of x_0.

For $t = 0, \ldots, T-1$

1. *Prediction*: evaluate for all possible values of x_{t+1} the *predictive distribution*

$$p(x_{t+1}|y_{1:t}) = \sum_{x_t} p(x_{t+1}|x_t) p(x_t|y_{1:t}) \tag{4}$$

2. *Filtering*: take into account the observation y_t for all possible values of x_{t+1} and compute the *filtering distribution*

$$p(x_{t+1}|y_{1:t+1}) = \frac{p(y_{t+1}|x_{t+1}) p(x_{t+1}|y_{1:t})}{\sum_{x_{t+1}} p(y_t|x_{t+1}) p(x_{t+1}|y_{1:t})} \tag{5}$$

■

Backward fixed interval smoothing

For $t = T-1, \ldots, 1$

$$p(x_t|y_{1:T}) = \sum_{x_{t+1}} \frac{p\left(x_{t+1}|x_t\right) p\left(x_t|y_{1:t}\right)}{\sum_{x_t} p\left(x_{t+1}|x_t\right) p\left(x_t|y_{1:t}\right)} p(x_{t+1}|y_{1:T}) \tag{6}$$

■

Although simple algebraically, these recursions are computationally very intensive in practice.

- There are n_s^{mL} possible values for x_{t+1}. Therefore the evaluation of Eq. (5) requires the computation of n_s^{mL} values of the probability. For each of the values of x_{t+1} there are n_s terms in the sum (indeed the case considered is a special case as x_{t+1} and x_t share $L-1$ components).

- The filtering step in Eq. (5) requires the evaluation of n_s^{mL} products, and the common normalizing constant requires the evaluation of a sum of n_s^{mL} terms.
- In the backward recursion (Eq. (6)) there are n_s^{mL} terms to be considered (x_t can take n_s^{mL} different values). The normalizing constant also contains n_s^{mL} that need to be summed.

As a conclusion for this section, we realize that the a direct implementation of the forward-backward algorithm leads to very difficult combinatorial problem as soon as n_s^{mL} is large. We therefore propose here a numerical approximation of all these quantities based on particle filter techniques.

4 Particle filtering approximation

Particle filtering methods rely on the principle of Monte Carlo methods, which we briefly recall here. Monte Carlo methods are very efficient at estimating integrals or sums that contain many terms.

4.1 The Monte Carlo method

Monte Carlo methods have proved to be very efficient at tackling such complex problems. They have been successfully applied in physics for 50 years, image processing for nearly 20 years and statistics for over a decade where they have revolutionized Bayesian statistics. The basic principle of Monte Carlo methods consists of replacing the algebraic representation of a distribution π, defined on some space X, by a population based representation. More precisely assume that we know how to produce N samples, the population, distributed according to π, then the probability of any region A of X, *i.e.* $\int_A \pi(x)\,dx$, can be approximated by the number of samples that belong to A. Now if we wish to approximate an integral of the form

$$I(f) = \int_{\mathsf{X}} f(x)\,\pi(x)\,dx$$

(where here $\int$ either means discrete or continuous sum), then a Monte Carlo estimator of $I(f)$ is given by

$$\widehat{I}(f) = \frac{1}{N}\sum_{i=1}^{N} f(x_i).$$

Intuitively this estimator ought to be efficient, as the samples (x_i) tend to concentrate on regions of high probability (*i.e.* where information is) and avoid regions of low probability, therefore making the most of the available computational power. This statement can be made mathematically rigorous, and it can be proved that under fairly general conditions, the rate of convergence of this estimator to the true value of the integral is of the order $O\left(\frac{1}{\sqrt{N}}\right)$, that is the rate of convergence is *independent of the dimension* of X.

4.2 Description of the algorithm

In the light of the previous introductory subsection, it seems legitimate to approximate at each time instant t all the sums required in the forward-backward algorithm by a sum over a subset of $N << n_s^{mL}$ adequately chosen symbol sequences. The particle filter algorithm can be thought of as being as a way of building a subset of the set of all possible symbol sequences that naturally focuses on the area of interest (*i.e.* according to their probabilities). Then computation are performed on this reduced grid, therefore saving considerable amounts of resources. N is therefore a parameter that needs to be chosen by the user, which will be a compromise between computational power available and the precision required to recover the symbol sequence accurately.

Here we present an algorithm that takes advantage of the fact that the number of possible states is finite. As discussed later in this section, it is possible to further reduce the complexity of our algorithm by introducing another degree of numerical approximation. We introduce here the notation $x_t^{(i)}$ for the i^{th} particle (or sample) of the system at time t. $x_t^{(i)}$ consists of a string of symbols $(s_t^{(i)}, \ldots, s_{t-L+1}^{(i)})$. Again the proposed algorithm is a numerical approximation of the forward-backward algorithm presented in the previous section. We start here with the prediction/filtering steps described in Eq. (4)-(5).

Modified Forward filtering via particle filter

- Initialization: For $i = 1, \ldots, N$ sample $x_0^{(i)} \sim p(x_0)$.
- At time $t+1$ we observe y_{t+1}.
  For $i = 1, \ldots, N$
  (a) Consider the n_s^m possible extensions $(\widetilde{x}_{t+1}^{(i,j)}, 1 \leq j \leq n_s^m)$ of $x_t^{(i)}$.
  (b) Compute their weights *i.e.* for $j = 1, \ldots, n_s^m$

$$\widetilde{w}_{t+1}^{(i,j)} \propto \phi\left(p(y_{t+1}|\widetilde{x}_{t+1}^{(i,j)}), p(\widetilde{x}_{t+1}^{(i,j)}|x_t^{(i)})\right) w_t^{(i)},$$

where $\widetilde{x}_{t+1}^{(i,j)} = (a_t, [x_t^{(i)}]_{2:L})$ and $\phi : \mathbb{R}_+^2 \to \mathbb{R}_+$. These weights are normalized, so that

$$\sum_{i=1}^{N}\sum_{j=1}^{n_s^m} \widetilde{w}_{t+1}^{(i,j)} = 1.$$

Select N particles among the $N \times n_s^{mL}$ present particles, according to their weights, using residual sampling for example.

Set

$$w_{t+1}^{(i)} \propto \frac{p(y_{t+1}|x_{t+1}^{(i)})p(x_{t+1}^{(i)}|x_t)}{\phi\left(p(y_{t+1}|x_{t+1}^{(i)}), p(x_{t+1}^{(i)}|x_t)\right)}.$$

■

In our implementation we have chosen $\phi(u,v) = u^\alpha(v+\beta)^\alpha$ with $\alpha \in (0,1]$ and $\beta > 0$, therefore defining

$$\widetilde{w}_{t+1}^{(i,j)} \propto p^\alpha(y_{t+1}|x_{t+1}^{(i,j)})\left[p(x_{t+1}^{(i,j)}|x_t) + \beta\right]^\alpha.$$

This modification of the algorithm proves to be beneficial in practice. Indeed the factor α allows for the preservation of diversity by tempering the selectivity of the selection step. This is of interest in particular when L is large and introduces memory in the system. The factor β proves to be of practical interest as, due to the numerical approximation introduced, the decoder tends to produce marginal posterior priors for the symbols that can be equal to 0, which is not desirable when using an exploration strategy as we do. As we shall see, the introduced discrepancy can be corrected in the backward step of the algorithm.

This forward recursion produces, for each time $t = 1, \ldots, T$, a set of couples (symbol,weights) $((s_t^{(i)}, w_t^{(i)}), i = 1, \ldots, N)$ from which one can represent the filtering distributions $(p(s_t|y_{1:t}), 1 \leq t \leq T)$. The state space is now paved with a marked grid of symbols which can represent $(p(s_t|y_{1:t}), 1 \leq t \leq T)$ and we now work on this $T \times N$ grid : the exponential complexity has disappeared, and the cost of the algorithm is now controlled by the parameter N.

Note that it might be that n_s^m is still too large for our particular application. If needed it is possible to develope a "local" MCMC algorithms that preselects a predefined number $N_e << n_s^m$ (whose value depends on our computational budget) of possible extensions for $x_t^{(i)}$. This is part of the possible extensions of the algorithm, which we do not consider here.

We complete the algorithm by describing the backward recursion corresponding to Eq. (6). It consists of drawing, say M, possible strings of symbols $(s_{1:T}^{1:m}(k), 1 \leq k \leq M)$, that are distributed according to the joint distribution $p(s_{1:T}^{1:m}|y)$. Then Monte Carlo estimators for the marginal posterior distributions $(p(s_t|y), 1 \leq t \leq T)$ are given by the following sums

$$\widehat{p(s_t = a_k|y)} = \frac{1}{M}\sum_{k=1}^{M} \mathbb{I}_{a_k}(s_t(k)),$$

where $\mathbb{I}_a(s) = 1$ if $s = a$ and 0 otherwise. The algorithm proceeds as follows,

Backward smoothing on the "filter grid"

For $k = 1, \ldots, M$

(a) $x_T(k) \sim P(x_T^{(i)} = x) = w_T^{(i)}$.

- For $t = T-1, \ldots, 1$

  $x_t(k)|x_{t+1}(k) \sim \mathbb{P}(x_t(k) = x_t^{(i)}|x_{t+1}(k+1)) = w_{t|t+1}^{(i)}$ where

  $$w_{t|t+1}^{(i)} \propto p(x_{t+1}(k)|x_t^{(i)})w_t^{(i)},$$

  are normalized weights.

■

The complete algorithm (see Fig. 2) now consists of alternating the estimation of the parameters $\theta = (h, \sigma^2)$ and the sequence of symbols followed by an iteration of the decoder, resulting in an EM-like algorithm. This is classic and not detailed here. Another possibility, not explored here, consists of embed the estimation of h and σ^2 in the particle filter/smoother algorithm above, possibly using a state space representation (h, σ^2) which allows one to take non-stationarity into account.

5 Numerical simulation

We consider here a numerical simulation where $A = \{+1, -1\}$ with $m = n = 2$, $L = 10$ and $T = 128$ with 4 pilot symbols in order to help the estimation of h and σ^2. Results are presented in Fig. 3 for four iterations of the decoder. Further results and comparison with other suboptimal methods are presented in [1] and demonstrate the efficiency of the procedure in terms of performance/complexity.

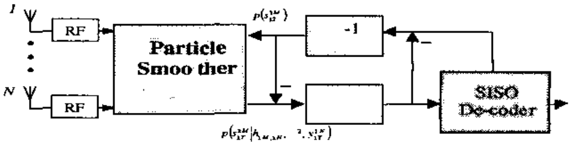

Figure 2: Schematic of the receiver.

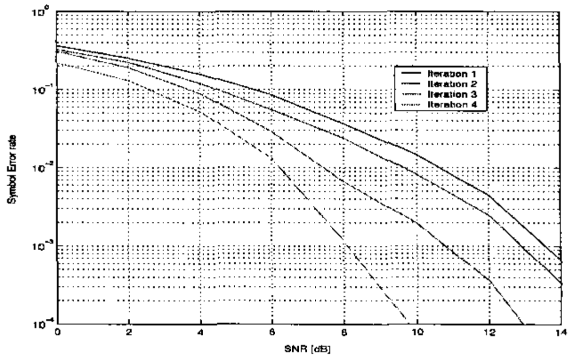

Figure 3: BER as a function of the SNR.

6 REFERENCES

[1] Christophe Andrieu, Robert J. Piechocki and Joe P. McGeehan, "Particle Smoothing Techniques with Turbo Principle for MIMO Systems," *submitted IEEE Trans. Comms.*.

[2] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. on Inf. Theory*, pp. 284-287, Mar. 1974.

[3] A. Doucet, N. de Freitas, and N. Gordon, *Sequential Monte Carlo Methods in Practice*, Statistics for Engineering and Information Science. Springer Verlag, New York, Heidelberg, Berlin, 2001.

[4] W. Fong, SJ. Godsill, A. Doucet and M. West, "Monte Carlo Smoothing with Application to Audio Signal Enhancement", *IEEE Trans. Signal Processing*, vol. 50, no. 2, pp. 438-448, 2002.

[5] E. Punskaya, C. Andrieu, A. Doucet and W.J. Fitzgerald, "Particle Filtering for Demodulation in Fading Channels with Non-Gaussian Additive Noise", *IEEE Trans. Com.*, vol. 49, no. 4, pp. 579-582, 2001.

[6] J.B. Anderson and S. Mohan, "Sequential coding algorithm: a survey and cost analysis,", *IEEE Trans. on Com.*, vol. 32, pp. 169-176, 1984.

[7] S.J. Simmons, "Breadth-first trellis decoding with adaptive effort," *IEEE Trans. on Com.*, vol. 38, pp. 3-12, 1990.