

JOINT IDENTIFICATION OF MERGE SIGNALS IN NON-  
COOPERATIVE DIGITAL TELECOMMUNICATIONS

5 The present invention relates generally to the methods and systems for processing the signal in digital telecommunications, and more particularly to the methods and systems for blind separation of such signals or signals of similar characteristics.

10 In digital telecommunications, a transmitter seeks to transmit a series of information bits to one (or several) receiver(s). The transmitter performs a series of coding operations in order to make it possible to transmit the information over a physical medium that is  
15 also called propagation channel (such as an optical fiber, or as is more particularly the case here, by unconstrained propagation in space). This propagation channel is not, generally, perfect in as much as the received signal is not an exact replica of the  
20 transmitted signal (thermal disturbances, reverberations, dispersion).

At the transmitter, a constellation (ASK, M-PSK, M-QAM) is provided which is a finite set of symbols in the  
25 complex plane and a function which associates a symbol of the constellation with a defined number of consecutive bits.

Each symbol of the series is multiplied by a continuous  
30 wave form (or forming filter), shifted in time to form the baseband signal.

This baseband signal is transposed to a certain frequency, called carrier frequency, then transmitted  
35 by a radio interface (an antenna in the case of a wireless transmission).

On reception, the reverse operations are performed to reconstitute the information bits from the signal measured on the radio interface. In particular, the demodulation is the operation which identifies, on  
5 reception, the symbols actually transmitted by the transmitter from the received signal.

The term blind demodulation applies when one of the parameters of the transmission is unknown to the  
10 transmitter. In cooperative telecommunications, most of the parameters are defined by agreement between the transmitter and the receiver (carrier frequency, constellation type, time shift between the waveforms, transmission power). The only unknowns which remain are  
15 possible synchronization errors and the effect of the propagation channel. These parameters are, in general, estimated by regular transmission sequences, called pilots, known to both parties. The operation compensating the effects of the channel before  
20 demodulation is called equalization.

In non-cooperative telecommunications however, all of the parameters are unknown - or known in order-of-magnitude terms only - and there are no pilot sequences  
25 to facilitate the equalization.

The separation of sources is a generic problem in which the aim is to extract, from a composite signal, the source signals of which it is composed.

30 Blind separation of sources designates the subclass of the separation problems when the characteristics of the source signals and their mix are unknown or very partially known a priori. Statistical assumptions on  
35 the source signals and/or on the nature of their mix prove necessary for this problem to be able to be solved. It is, for example, commonly assumed:

- that the sources are mutually independent;
- that the maximum number of constituent sources is known;
- that the mix is linear;
- 5 - that there are discriminating statistics between the constituent sources.

Our focus here is on the blind separation and demodulation of digital telecommunications signals in a  
10 non-cooperative context. Our interest is more specifically focused on a case where there are no clear discriminating statistics between the sources.

An example that can be cited is the identification of  
15 signals transmitted according to the "DoubleTalk® Carrier-in-Carrier®" satellite transmission protocol. This satellite transmission technique uses one and the same frequency band for the continuous exchange of information between two pairs. An intercepted  
20 DoubleTalk signal therefore consists of the superimposition in time and in frequency of two signals of the same characteristics. The identification of these signals is a problem of blind separation and demodulation of sources.

25

Out of the blind demodulation methods in digital telecommunications, interest is focused on those which rely:

- 30 - on the EM (Expectation-Maximization) classification algorithm for the estimation of the parameters of the signal; and
- on the Viterbi algorithm for the demodulation.

35 The documents (S. Barembrouch, "*Méthodes approchées de maximum de vraisemblances pour la classification et identification aveugles en communications numériques*", thesis report, Telecom ParisTech, 7 March 2011) and

(E. Punskeya, "Sequential Monte Carlo methods for digital communications", thesis report, Cambridge University, 2003) disclose two similar methods allying the particle filtering with the EM and Viterbi algorithms.

In these two documents, the model of the signal to be demodulated is expressed as follows:

$$y(i) = \sum_{j=0}^J h(j)s(i-j) + \eta(i)$$

or else:

$$y(i) = \langle h | S(i) \rangle + \eta(i)$$

where  $S(i)$  is the vectorized version of the symbols  $s(i-J)$  to  $s(i)$  which belong to a constellation  $C$ ,  $h$  is a linear filter modelling the propagation channel, and  $\eta$  is a Gaussian white noise of variance  $\sigma^2$ . The EM algorithm makes it possible to estimate the filter  $h$  and the variance  $\sigma^2$  according to a probabilistic paradigm and the Viterbi algorithm then makes it possible to reconstitute the sequence  $\hat{S}$  of vectorized symbols which maximizes the probability of the observations of the composite signal  $y$ .

The EM algorithm makes it possible to iteratively approximate the value of the vector of parameters  $\theta = (h, \sigma^2)$  which maximizes the likelihood of the observed signal. On each iteration, a step E (for expectation) and a step M (for maximization) are applied in succession.

On the iteration  $k$ ,  $\theta^{(k)} = (h^{(k)}, \sigma^{2(k)})$  of the parameters of the system have been estimated.

- Step E: this step seeks to test the conditional expectation of the log-likelihood of the observations:

$$L_{\theta^{(k)}}(\theta) = E\{\log P_{\theta}((y(1:I), S(1:I) = X(1:I)) | X(1:I) \sim P_{\theta^{(k)}}(\cdot | y(1:I)))\}$$

which is rewritten

$$L_{\theta^{(k)}}(\theta) = \sum_i \sum_{X \in C^J} \beta_{i,\theta^{(k)}}(X) \cdot L'_{\theta}(y(i)|X(i) = X) (+cste(\theta))$$

With, for any time,  $i$ ,  $X(i)$  is the vector variable in  $C^J$  which denotes a possible value for the vector of symbols  $S(i)$  actually transmitted, the writing  $y(1:I)$  denotes the series of observations  $(y(1), \dots, y(I))$  of the composite signal  $y$ ,  $\beta_{i,\theta^{(k)}}(X) = P_{\theta^{(k)}}(X(i) = X | y(1:I))$  is the smoothing distribution and  $L'_{\theta}(y(i)|X(i) = X) = \log P_{\theta}(y(i)|X(i) = X) = |y(i) - \langle h|X \rangle|^2 / 2\sigma^2$ . The term  $cste$  does not depend on  $\theta$ . It does not therefore have to be calculated, and is omitted in the definition of the conditional expectation of the log-likelihood  $L_{\theta^{(k)}}$ .

The issue of the step E is to calculate the smoothing distributions  $\beta_{i,\theta^{(k)}}(X)$ , which is done by successively calculating the so-called filtering  $(P_{\theta^{(k)}}(X(i) = X | y(1:i)))$  then smoothing  $(\beta_{i,\theta^{(k)}}(X))$ , distributions, by means for example of the Baum and Welch Forward-Backward algorithm.

- Step M: once the quantities  $\beta_{i,\theta^{(k)}}(X)$  are calculated, the step M updates the parameters  $\theta^{(k)}$  as  $\theta^{(k+1)}$  according to

$$\theta^{(k+1)} = \operatorname{argmax}_{\theta} L_{\theta^{(k)}}(\theta)$$

In theory, the series of vectors  $\theta^{(k)}$  converges toward the vector  $\theta$  which maximizes the likelihood of the observations. However, applied to the demodulation of a digital communication signal, the EM algorithm is  
 5 faulted in computational complexity terms fairly rapidly with the size of the filter  $h$  and/or the size of the constellation  $C$ .

The contribution of the two documents cited above is to  
 10 go beyond this complexity limitation with the notions of particle filtering and smoothing which make it possible to approximate the calculation of the smoothing distribution  $\beta_{i,\theta^{(k)}}(X)$  by limiting the number of states  $X$  to be evaluated.

15 Thus, they consider that  $\beta_{i,\theta^{(k)}}(X) \simeq \sum_{r \leq R} \omega_{r,i} \cdot \delta(X - Z_{r,i})$  where  $\delta(\cdot)$  is the Dirac distribution over the state space  $C^J$ ; for any  $i$ , the  $(Z_{r,i})_{1 \leq r \leq R}$  are  $R$  representatives of the state space that are well chosen and in small number  
 20 compared to the size of this space; for any  $i, r$ ,  $\omega_{r,i} \in [0,1]$  and for any  $i$ ,  $\sum_r \omega_{r,i} = 1$ . The sets  $\{(Z_{r,i}, \omega_{r,i})\}_{i,r}$  are called particles and the issue of the particle method is to formulate an algorithm making it possible to calculate them recursively on the time index  $i$ .

25 For this, the following two steps are carried out:

- first of all, in a so-called filtering step, the filtering distribution  $P_{\theta^{(k)}}(X(i) = X | y(1), \dots, y(i))$  is  
 30 calculated for any  $i$ , in the form

$$P_{\theta^{(k)}}(X(i) = X | y(1), \dots, y(i)) \simeq \sum_{r \leq R} \xi_{r,i} \cdot \delta(X - Z_{r,i})$$

with, for any  $i, r$ ,  $\xi_{r,i} \in [0,1]$  and for any  $i$ ,  $\sum_r \xi_{r,i} = 1$ ;

35

- then, a so-called smoothing step in which the coefficients  $\xi_{r,i}$  are transformed into coefficients  $\omega_{r,i}$  which incorporate the observations after the time  $i$  to obtain

5

$$\beta_{i,\theta^{(k)}}(X) = \mathbf{P}_{\theta^{(k)}}(X(i) = X | y(1), \dots, y(I)) \simeq \sum_{r \leq R} \omega_{r,i} \cdot \delta(X - Z_{r,i})$$

We take a number called threshold and a transition kernel  $\mathbf{A}_i(\cdot, \cdot): \mathcal{C}^J \times \mathcal{C}^J \rightarrow [0,1]$  which depends possibly on the sample  $y(i)$ ; the calculation of the filtering distribution  $\mathbf{P}_{\theta^{(k)}}(X(i) = X | y(1), \dots, y(i))$  is performed recursively, according to the steps:

- as input, let  $\left\{ \left( (Z_{r,i-1}, \xi_{r,i-1}) \right) \right\}_r$  be a particle approximation of  $\mathbf{P}_{\theta^{(k)}}(X(i-1) = X | y(1), \dots, y(i-1))$ ;
- if  $\sum_r \xi_{i-1,r}^2 \leq \text{threshold}$ 
  - o assume for any  $r$ ,  $\dot{Z}_{i-1,r} = Z_{i-1,r}$  and  $\dot{\xi}_{i-1,r} = \xi_{i-1,r}$
- if  $\sum_r \xi_{i-1,r}^2 > \text{threshold}$ 
  - o draw a multinomial variable:  $(Y_1, \dots, Y_R) \sim \text{Mult}_R(\xi_{i-1,1}, \dots, \xi_{i-1,R})$  (the  $Y_r$  are integer random variables whose sum has the value  $R$  and the expectation of each variable has the value  $R \cdot \xi_{i-1,r}$ )
  - o assume  $\epsilon: \llbracket 1, R \rrbracket \rightarrow \llbracket 1, R \rrbracket$ ,  $\epsilon(r) = \inf \{r', \sum_{1 \leq r' \leq r} Y_{r'} \geq r\}$
  - o for any  $r \in \llbracket 1, R \rrbracket$ ,  $\dot{Z}_{i-1,r} = Z_{i-1,\epsilon(r)}$  et  $\dot{\xi}_{i-1,r} = 1/R$  and  $\dot{\xi}_{i-1,r} = 1/R$

30

25

20

15

- for each  $1 \leq r \leq R$ :
  - o draw  $Z_{i,r}$  according to  $A_i(\cdot, \dot{Z}_{i-1,r})$
  - 5 o assume:  $\Xi_{i,r} = \xi_{i-1,r} \cdot (P_{\theta^{(k)}}(Z_{i,r} | y(i), \dot{Z}_{i-1,r}) / A_i(Z_{i,r}, \dot{Z}_{i-1,r}))$
  - o normalize the weights:
    - o  $\Xi_i = \sum_r \Xi_{i,r}$
    - o  $\xi_{ij} = \Xi_{i,r} / \Xi_i$

10

As output,  $\{(Z_{i,r}, \xi_{i,r})\}_r$  is a particle approximation of  $P_{\theta^{(k)}}(X(i) = X | y(1), \dots, y(i))$ .

The success of this method depends on the choice of a  
 15 kernel  $A_i(\cdot, \cdot)$  suited to the particular features of the problem being processed.

The documents cited propose and compare different methods for deducing smoothing coefficients from the  
 20 filtering coefficients. We retain in particular the method called "Fixed Lag Smoothing".

After the estimation of the parameters of the model of the composite signal by this EM-particle algorithm, the  
 25 prior arts cited above use an adaptation of the Viterbi algorithm to produce the demodulation. This adaptation - called stochastic Viterbi - limits the exploration of the lattice of the signal to the states of the particles obtained upon the last iteration of the EM  
 30 algorithm.

This approach thus makes it possible to solve the problem of the digital demodulation with a complexity applicable to filters and/or constellations of large  
 35 size.

These prior arts present a number of limitations for being directly adapted to the problem of blind separation and demodulation of merged digital signals because, among other things,

5

- their formalization is limited to a single source;
- they assume that a synchronization is acquired between the transmitter and the receiver. However, it is impossible for an interceptor to synchronize on several signals at the same time. It follows therefrom that, in the case of a mix of signals, it can only be assumed that one synchronization is acquired, thus compromising the operation of the purely sequential model of the methods cited;

10

15

- they consider a propagation channel causal. However, the synchronization fault induces on each channel (that is to say on each signal of the mix) an equivalent non-causal filter.

20

Furthermore, a problem specific to the example of the "DoubleTalk® Carrier-in-Carrier®" digital communications cited above is that the signals to be separated are, theoretically, transmitted at the same frequency, at the same instants and with the same bit rate. However, in practice, the time-frequency characteristics of these signals can differ slightly because of an imperfect synchronization of the transmitters of these signals. The operating point of this system relies on synchronization parameters that are very close (but not strictly identical) which are, in this case, unknown or at best known by their order of magnitude. It follows therefrom that it is impossible to claim to separate, in all particular cases, the signals by the conventional time-frequency filtering techniques or by disregarding one of the two signals in favor of the other.

25

30

35

One object of the present invention is to propose methods for blind separation and demodulation in digital telecommunications of signals included in a composite signal.

5

Another object of the present invention is to propose methods that make it possible to separate and demodulate digital signals blind that works equally with signals of different parameters and with signals of very close, even identical, modulation parameters.

10

Another object of the present invention is to propose methods that make it possible to separate and demodulate digital signals blind in real time, that is to say with complexities in processing time and in data storage that are linear according to the number of samples of the composite signal taken.

15

Another object of the present invention is to propose methods that make it possible to separate and demodulate two digitally stable digital signals blind.

20

Another object of the present invention is to propose methods that make it possible to separate and demodulate two digital signals blind with a channel and parameters that evolve over time.

25

Another object of the invention is the blind and joint demodulation of the different channels of a "DoubleTalk® Carrier-in-Carrier®" signal.

30

To these ends, the invention relates, according to a first aspect, to a real-time method for blind separation and demodulation of digital telecommunications signals, called channels, from the observation, by means of a single sensor, of a composite signal comprising these signals, the parameters of these channels including their modulation

35

type, their amplification, their phase-shift, their delay time at the sensor, their frequency and their symbol time, these parameters for the different channels being able to be different, substantially or  
5 perfectly equal, this method comprising the following steps:

- acquisition of a first plurality of observations of the composite signal performed by means of the  
10 single sensor;
- estimation, from the first plurality of observations acquired, of the parameters of the channels in terms of the "Maximum Likelihood" by an Expectation-Maximization algorithm, the  
15 conditional expectation of the log-likelihood being calculated, in this algorithm, recursively by a particle filtering-smoothing method;
- joint demodulation of the channels according to a stochastic Viterbi algorithm.

20

The particle filtering-smoothing method comprises a particle approximation of the filtering distribution, this particle approximation assigning, in addition to weights, multiplicities to the particles of this  
25 approximation, a multiplicity being the number of particles representing a same state.

The particle filtering-smoothing method comprises a particle approximation of the smoothing distribution,  
30 this particle approximation assigning, in addition to weights, multiplicities to the particles of this approximation, a multiplicity being the number of particles representing a same state.

35 This method further comprises,

- a step of acquisition of a second plurality of observations of the composite signal performed by means of the single sensor;
- a step of estimation, from the second plurality of observations acquired, of the parameters of the channels in terms of the Maximum Likelihood by an Expectation-Maximization algorithm, the conditional expectation of the log-likelihood being calculated, in this algorithm, recursively by a particle filtering-smoothing method;
- a step of long-term estimation of the parameters, this long-term estimation of the parameters being associated with the second plurality of observations and being a linear combination of the parameters estimated from the first plurality of observations and of the estimated parameters of the second plurality of observations.

The invention relates, according to a second aspect, to a computer program product implemented on a memory medium, capable of being implemented in a computer processing unit and comprising instructions for the implementation of a method as presented above.

Other objects and advantages of the invention will become apparent in light of the description of embodiments, given below with reference to the attached drawings in which:

- figures 1-6 schematically illustrate different steps of a method for blind source separation according to various embodiments;
- figures 7-15 schematically illustrate the results obtained at different steps of the separation method applied to a composite "DoubleTalk@Carrier-in-Carrier®" signal.

In one embodiment, a composite signal  $y(t)$  comprising a mix of  $N$  source signals  $(s_n(t))_{n=1..N}$  is recorded by means of a single sensor. The signals  $(s_n(t))_{n=1..N}$ , hereinafter called channels, contain the information symbols  $(s_n(i))_{i=1..G_n}$  deriving respectively from the constellations  $C_n = (c_g)_{g=1..G_n}$  and carried respectively by wave functions  $h_n(\cdot)$ . The composite signal  $y(t)$  is in theory written:

$$y(t) = \sum_{n=1}^N A_n \cdot e^{i2\pi\delta f_n t} \left( \sum_{j=-\infty}^{j=+\infty} h_n(t - jT_n - \tau_n) s_n(j) \right) + \eta(t)$$

10

in which:

- $A_n$  is the complex gain of the source signal  $s_n$  (that is to say,  $A_n = |A_n|e^{i2\pi\varphi_n}$  with  $|A_n|$  the amplification (or gain) and  $\varphi_n$  the phase-shift (or phase error) on the channel  $n$ );
- $h_n$  is the analytical form of the forming filter of the channel  $n$  (for example, a Raised Cosine filter);
- 20 -  $s_n$  is the series of symbols of the constellation  $C_n$  transmitted over the channel  $n$  and  $s_n(i)$  is the  $i$ th symbol transmitted over the channel  $n$ ;
- $\delta f_n$  is the carrier residue (or, more generally, the frequency) on the channel  $n$ . In particular, these residues can be substantially equal;
- 25 -  $T_n$  is the symbol time on the channel  $n$ . In particular, these symbol times can be substantially equal ( $T_1 \simeq T_2 \simeq \dots \simeq T_N$ );
- $\tau_n$  is the delay time on the channel  $n$  at the single sensor. These times are chosen, with no loss of generality, between  $-T_n$  and 0;
- 30 -  $\eta(\cdot)$  is a random disturbance.

The composite signal  $y(t)$  is sampled with the period  $T_e$ :

35

$$y(i) \triangleq \mathbf{y}(iT_e) = \sum_{n=1}^N A_n \cdot e^{i2\pi\delta f_n T_e i} \left( \sum_{j=-\infty}^{j=+\infty} h_n(iT_e - jT_n - \tau_n) s_n(j) \right) + \eta(i)$$

$T_e$  is chosen to be less than twice the symbol times  $T_n$ , that is to say that, for any  $1 \leq n \leq N$ ,  $T_e < 2T_n$ . The digital noise  $\eta$  is a white, Gaussian noise of variance  $\sigma^2$ .

The set of the parameters of the model of the composite signal  $y(t)$  above is denoted in the form of a parameter vector  $\theta = ([A_n, T_n, \delta f_n, \tau_n]_{n=1..N}, \sigma^2)$ .

For each channel  $n$ , the  $J$  symbols directly following  $i$  and the  $J+1$  symbols directly prior to or concomitant with  $i$  are considered. For any channel  $n$ , the vector consisting of these symbols is denoted  $S_n(i) = [s_{n,j}(i)]_{j=1..2J+1} \in C_n^{2J+1}$  and the grouping of these vectors is denoted  $[S_n(i)]_n \in C_1^{2J+1} \times \dots \times C_N^{2J+1}$ .

The time difference between the  $i$ th sample intercepted and the  $j$ th component of the vector  $X_n(i)$  is denoted  $t_n(i, j) (= t_{n,ij}(T_n, \tau_n))$ .

The signal  $y$  is approximated by truncating the wave functions according to the equation

$$y(i) \simeq \sum_{n=1}^N A_n \cdot e^{i2\pi\delta f_n T_e i} \left( \sum_{j=1}^{j=2J+1} h_n(t_n(i, j)) s_{n,j}(i) \right) + \eta(i).$$

An order relationship denoted  $\prec_n$  is added to each constellation  $C_n$ .

$C_1^{2J+1} \times \dots \times C_N^{2J+1}$  to  $(C_1 \times \dots \times C_N)^{2J+1}$  is identified via the bijection  $O: C_1^{2J+1} \times \dots \times C_N^{2J+1} \rightarrow (C_1 \times \dots \times C_N)^{2J+1}, [X_n]_n \rightarrow$

$\left( (o_{1,1}([X_n]), \dots, o_{1,N}([X_n])), \dots, (o_{2J+1,1}([X_n]), \dots, o_{2J+1,n}([X_n])) \right)$ , with,  
for any  $n, X_n = [x_{n,j}]_{j=1 \dots 2J+1}$  and  $o_{j,n'}([X_n]_n) = x_{n',j}$ .

The order relationship  $<$  is defined on  $C_1^{2J+1} \times \dots \times C_N^{2J+1}$  by:  
5 for  $U_1 \in C_1^{2J+1} \times \dots \times C_N^{2J+1}, U_2 \in C_1^{2J+1} \times \dots \times C_N^{2J+1}$ :

$$U_1 < U_2 \text{ (} j_0, n_0 \text{), } \forall j < j_0, \forall n, o_{j,n}(U_1) = o_{j,n}(U_2) \text{ and } \forall n < n_0, o_{j_0,n}(U_2) \\ = o_{j_0,n} \text{ and } o_{j_0,n_0}(U_1) <_{n_0} o_{j_0,n_0}(U_2)$$

10 The separation and the demodulation of the source signals  $(s_n)_{n=1..N}$  aims to extract the information symbols  $(s_n(j))_n$  of each signal  $(s_n)_{n=1..N}$  from the composite signal  $y(i)$  observed on a sensor.

15 Referring to figure 1, samples (or even observations)  $y(i)$  of the composite signal  $y$  are supplied as input to the blind separation system 10.

The blind separation system 10 extracts, from the  
20 observations  $y(i)$  of the composite signal, the information symbols  $(\hat{s}_n(t))_{n=1..N}$  of each of the  $N$  signals  $(s_n)_{n=1..N}$  included in the composite signal  $y$ .

This estimation is made in terms of the Maximum  
25 Likelihood; that is to say that the blind separation system 10 minimizes the following function:

$$\log P_\theta(y(1:I), s_1(1:I), \dots, s_N(1:I))$$

30 Depending on the parameter vector  $\theta = ([A_n, T_n, \delta f_n, \tau_n]_{n=1..N}, \sigma^2)$  and the series  $(s_n(1:I))_{n=1..N}$  (where  $I$  is a natural integer greater than or equal to one), this operation is performed, among others, by means of an EM algorithm

performed on particle approximations of the expectation of the log-likelihood.

Now referring to figure 2, the blind separation system  
5 10 comprises

- the preprocessing module 1 configured to filter, downsample and subdivide, the samples  $y(i)$  of the composite signal  $y$ , into frames  $y_1, \dots, y_m, \dots, y_M$  of length  $I$ . It follows therefrom that the frames  $y_1, \dots, y_m, \dots, y_M$  are successively obtained at the output of the preprocessing module 1. This preprocessing module 1 can also produce a pre-estimation  $\theta_m^{(0)}$  of the parameter vector  $\theta$  when it is applied to the frame  $y_1$ ;
- the modules 2-4 are configured to estimate, from each frame  $y_m$ , the information symbol  $(\hat{s}_{n,m})_{n=1..N}$  of the signal  $(s_n)_{n=1..N}$  transmitted for the duration of the frame  $y_m$ .

20

The module 2 implements the EM algorithm by particle filtering and smoothing. This module 2 operates on  $K$  iterations per frame  $y_m$  and gives as output a parameter vector  $\theta_m^{(K)} = ([A_{nm}^{(k)}, T_{nm}^{(k)}, \delta f_{nm}^{(k)}, \tau_{nm}^{(k)}]_{n=1..N}, \sigma_m^{2(k)})$  estimated on the frame  $y_m$  of the composite signal  $y$ . This module 2 requires the definition of two additional parameters:

- $\theta_m^{(0)}$  to initialize the recursive estimation of the parameter vector  $\theta$ ; and
- $\hat{p}_\alpha(m, 0)$  which corresponds to a "particle approximation" of the last sample of the signal directly preceding the first sample of the frame  $y_m$  (that is to say, the sample  $y_{m-1}(I)$ ,  $I$  being the number of samples per frame). In the case where  $m = 1$ , this approximation can, for example, result from a uniform draw of the possible states.

35

The module 3 performs the long-term recursive tracking of the parameters of the signal. This module 3 takes as input the vector  $\theta_m^{(K)}$  delivered by the module 2 for the frame  $m$  and a vector  $\theta_{m-1}$ , giving a long-term estimation of the parameters of the signal up to the frame  $m-1$ . This parameter  $\theta_{m-1}$  generally originates from this same module 3 in the frame  $m-1$ . The module 3 produces the long-term estimation of the parameters  $\theta_m$  by linearly combining  $\theta_m^{(K)}$  and  $\theta_{m-1}$ .

10

A demodulation module 4 implements a variant of the stochastic Viterbi algorithm from the current frame  $y_m$  of the composite signal and from the parameter vector  $\theta_m$  estimated over the long term by the module 3. This demodulation module 4 supplies as output an estimation  $\hat{s}_{1,m}, \dots, \hat{s}_{N,m}$  of the information symbols  $s_{1,m}, \dots, s_{N,m}$  transmitted over the duration of the frame  $y_m$  by each of the channels  $s_1, \dots, s_N$ .

20 The information symbols estimated  $\hat{s}_{1,m}, \dots, \hat{s}_{N,m}$  in the course of the frames  $y_1, \dots, y_M$ , respectively, on the channels,  $s_1, \dots, s_N$  are, respectively, loaded into buffer memories (see "buffer" in figure 2) to form, as output from the blind separation system 10, streams of parallel information symbols corresponding to the signals  $s_1, \dots, s_N$ .

Figure 3 details the module 2 for the implementation of the EM algorithm by particle filtering and smoothing and, in particular, the interactions between the step E implemented by the module 2.1 and the step M implemented by the module 2.2 in the course of the iterations  $k = 1, \dots, K$  on a frame  $y_m$ . The module 2 encapsulating the EM algorithm supplies as output an estimation of the parameter vector  $\theta_m^{(K)}$  on the frame  $y_m$  after  $K$  iterations from:

30

35

- the current frame  $y_m$ ;
- a set of particles denoted  $\hat{P}_\alpha(m, 0)$  approximating the signal with the sample preceding the first sample of the frame  $y_m$ ; and
- an initialization parameter vector  $\theta_m^{(0)}$  initializing the EM algorithm.

On the iteration  $k$ , the module 2.1 for the implementation of the step E calculates, from  $\hat{P}_\alpha(m, 0)$  and from the parameter vector  $\theta_m^{(k-1)}$  (that is to say the parameter vector obtained on the preceding iteration), a set of particles denoted  $P_{\tilde{\omega}}(m, 1:I)$  deriving from the smoothing distribution over all of the frame  $y_m$ . In the case where  $k = 1$  and  $m > 1$ ,  $\theta_m^{(k-1)} = \theta_{m-1}$  (given by the module 3 on the frame  $y_{m-1}$ ); if  $k = 1$  and  $m = 1$ ,  $\theta_m^{(k-1)} = \theta_0$  is an initialization parameter vector of the EM algorithm. Depending on the embodiments, it can be supplied by an operator, be chosen randomly or be a sub-product of the preprocessing module 1.

On the iteration  $k$ , the module 2.2 for the implementation of the step M refines the estimation of  $\theta_m^{(k)}$  as  $\theta_m^{(k+1)}$  from the frame  $y_m$ , from  $\theta_m^{(k)}$  and from the particles  $P_{\tilde{\omega}}(m, 1:I)$  obtained by the module 2.1 for the implementation of the step E (see figure 3).

In this regard, on the iteration  $k$ , the step E implemented by the module 2.1 evaluates the conditional expectation of the log-likelihood of the observations:

$$L_{\theta_m^{(k-1)}}(\theta) = \mathbf{Esp} \left\{ \log \mathbf{P}_{\theta_m} (y_m(1:I), [S_n(1:I) = X_n(1:I)]_n) \mid [X_n(1:I)]_n \sim \mathbf{P}_{\theta_m^{(k-1)}}(\cdot \mid y_m(1:I)) \right\}$$

which is rewritten

$$L_{\theta_m^{(k-1)}}(\theta) = \sum_i \sum_n \sum_{X_n(i) \in C_n^{2J+1}} \beta_{i, \theta_m^{(k-1)}}([X_n(i)]_n) \cdot L'_{\theta_m}(y_m(i) | [S_n(i) = X_n(i)]_n) (+L''(\theta_m))$$

in which:

$$\beta_{i, \theta_m^{(k-1)}}([X_n(i)]_n) = P_{\theta_m^{(k-1)}}([S_n(i) = X_n(i)]_n | y_m(1), \dots, y_m(I))$$

is the smoothing distribution and

$$L'_{\theta}(y_m(i) | [X_n(i) = X_n]_n) = \left| y_m(i) - \sum_{n=1}^N A_n e^{2\pi i \delta f_n T_n} \sum_{j=0}^{j=2J} x_{n,ij} h_n(t_{n,ij}(T_n, \tau_n)) \right|^2 / 2\sigma^2$$

10

$L''$  is a term not necessary to the variant of the step M implemented by the module 2.2. This term  $L''$  does not therefore need to be calculated. Hereinbelow, the index  $m$  of the frame  $y_m$  is understood to lighten the notations.

15

$L_{\theta^{(k)}}(\theta)$  is calculated by means of a particle approximation. A particle approximation is a parsimonious expression of the state space suited to the evaluation of  $L_{\theta^{(k)}}(\theta)$  in as much as the number of evaluation points is very small relative to the dimensions of the state space. Two types of particle approximations are considered.

20

According to a first, so-called single, particle approximation, a particle approximation of the filtering distribution  $P_{\theta^{(k)}}([X_n(i)]_n | y(1), \dots, y(i))$  is written

25

$$P_{\theta^{(k)}}([X_n(i)]_n | y(1), \dots, y(i)) \simeq \sum_r^R \xi_{r,i} \delta([X_n(i)]_n - [Z_{n,r}(i)]_n)$$

30

in which:

- $R$  is an integer called maximum number of particles given on initialization of this particle method;

- for any  $i$  and for any  $r \leq R$ ,  $\xi_{r,i}$  is a real number between 0 and 1, called filtering weight;
- for any  $i$ ,  $\sum_r \xi_{r,i} = 1$ ;
- for any  $n$ , any  $i$  and any  $r \leq R$ ,  $[Z_{n,r}(i)]_n = \left[ (z_{n,r,j}(i))_j \right]_n \in C_n^{2J+1}$  is a symbol vector called particle approximation support.

The particle filtering according to this first particle method is the recursive calculation of  $\mathbf{P}_{\theta^{(k)}}([X_n(i)]_n | y(1), \dots, y(i))$  for any  $i$  according to the approximation above.

Similarly, a single particle approximation of the smoothing distribution  $\beta_{i,\theta^{(k)}}([X_n(i)]_n) = \mathbf{P}_{\theta^{(k)}}([X_n(i)]_n | y(1), \dots, y(I))$  is an expression in the form:

$$\beta_{i,\theta^{(k)}}([X_n(i)]_n) \simeq \sum_r^R \omega_{r,i} \delta([X_n(i)]_n - [Z'_{n,r}(i)]_n)$$

in which:

- for any  $i$  and for any  $r \leq R$ ,  $\omega_{r,i}$  is a real number between 0 and 1 called smoothing weight;
- for any  $i$ ,  $\sum_r \omega_{r,i} = 1$ ;
- for any  $n$ , any  $i$  and any  $r \leq R$ ,  $[Z'_{n,r}(i)]_n = \left[ (z'_{n,r,j}(i))_j \right]_n \in C_n^{2J+1}$  is a symbol vector called particle approximation support.

This approximation can, in particular, be obtained from the particle approximation of the filtering distribution  $\mathbf{P}_{\theta^{(k)}}([X_n(i)]_n | y(1), \dots, y(i))$  above.

According to a second, so-called multiple, particle approximation, the particle approximation of the

filtering distribution  $\mathbf{P}_{\theta^{(k)}}([X_n(i)]_n | y(1), \dots, y(i))$  is an expression in the form

$$\mathbf{P}_{\theta^{(k)}}([X_n(i)]_n | y(1), \dots, y(i)) \simeq \sum_r^{R_{eff}(i)} \xi_{r,i} \cdot \mu_{r,i} \delta([X_n(i)]_n - [Z_{n,r}(i)]_n)$$

5

in which:

- $R$  is an integer called maximum number of particles given on initialization of this particle method;
- 10 - for any  $i$ ,  $R_{eff}(i) \leq R$  is an integer called effective number of particles on the generation  $i$ ;
- for any  $i$  and for any  $r \leq R_{eff}(i)$ ,  $\xi_{r,i}$  is a real number between 0 and 1 called filtering weight;
- for any  $i$  and for any  $r \leq R_{eff}(i)$ ,  $\mu_{r,i}$  is an integer
- 15 called multiplicity of the particle;
- for any  $i$ ,  $\sum_r \mu_{r,i} \xi_{r,i} = 1$ ;
- for any  $n$ , any  $i$  and any  $r \leq R_{eff}(i)$ ,  $[Z_{n,r}(i)]_n = \left[ (z_{n,r,j}(i))_j \right]_n \in C_n^{2J+1}$  is a symbol vector called particle approximation support.

20

The particle filtering according to this first particle method is the recursive calculation of  $\mathbf{P}_{\theta^{(k)}}([X_n(i)]_n | y(1), \dots, y(i))$  for any  $i$  according to the approximation above.

25

Similarly, a multiple particle approximation of the smoothing distribution  $\beta_{i,\theta^{(k)}}([X_n(i)]_n) = \mathbf{P}_{\theta^{(k)}}([X_n(i)]_n | y(1), \dots, y(I))$  is an expression in the form:

$$\beta_{i,\theta^{(k)}}([X_n(i)]_n) \simeq \sum_r^{R'_{eff}(i)} \omega_{r,i} \cdot v_{r,i} \delta([X_n(i)]_n - [Z'_{n,r}(i)]_n)$$

30

in which:

- for any  $i$  and for any  $r \leq R$ ,  $\omega_{r,i}$  is a real number between 0 and 1 called smoothing weight;
- for any  $i$  and for any  $r \leq R_{eff}(i)$ ,  $v_{r,i}$  is an integer called multiplicity of the particle;
- 5 - for any  $i$ ,  $\sum_r \omega_{r,i} v_{r,i} = 1$ ;
- for any  $n$ , any  $i$  and any  $r \leq R$ ,  $[Z'_{n,r}(i)]_n = \left[ (z'_{n,r,j}(i))_j \right]_n \in C_n^{2J+1}$  is a symbol vector called particle approximation support.

10 Two embodiments can be followed: according to the paradigm of the first particle approximation or according to the paradigm of the second (called multiple particle approximation) presented above.

15 Advantageously, the multiple particle approximation above of the filtering distribution, and that of the smoothing distribution, make it possible to eliminate the redundancy of the particles. This multiple approximation makes it possible, in effect, to also  
 20 omit the redundancy of the particles by gathering together the particles representing one and the same state. This multiple particle approximation, which makes it possible to limit the number of particles actually calculated and therefore to divide the  
 25 computation time and the memory space necessary to the execution of the algorithm, is detailed hereinbelow.

The following auxiliary variables are introduced:

- 30 - for any  $i$ , the application which, with a particle of the generation  $i$  designated by its index  $r$ , associates a particle of the generation  $i-1$ , called its ancestor  $\nabla_i(r)$ , is denoted  $\nabla_i: \llbracket 1, R_{eff}(i) \rrbracket \rightarrow \llbracket 1, R_{eff}(i-1) \rrbracket$ ;
- 35 - for any  $i$  and any channel  $n$ , we take a variable  $\pi_{n,i} \in \{0,1\}$  called jump indicator;

- for any  $i$  and any  $r \in \llbracket 1, R_{eff}(i) \rrbracket$ , an additional weight  $\alpha_{r,i} \in [0,1]$  is introduced called auxiliary weight, notably such that, for any  $i, \sum_{r=1}^{R_{eff}(i)} \alpha_{r,i} \cdot \mu_{r,i} = 1$ .

5

The precise definition of these auxiliary quantities will be done by recurrence in the complete specification of the procedure of the method for the blind separation of the signals.

10

To ensure the stability of the calculations, the auxiliary weights and the filtering weights are replaced, respectively, by their logarithms denoted  $L\alpha_{r,i}$  and  $L\xi_{r,i}$ .

15

The time differences  $t_n(i,j)$  between the sample  $i$  of the composite signal  $y$  and the  $j$ th sample of the symbol vector  $S_n(i)$  contributing to  $y_m(i)$  are calculated at the same time as the particles. The jump indicators  $\pi_{n,i} \in \{0,1\}$  has the value 1 if it is necessary to consider the arrival of a new symbol between the vectors  $S_n(i-1)$  and  $S_n(i)$  and 0 otherwise.

20

The sets are denoted:

25

- $$P_{\alpha}^{(k)}(m,i) = \left( R_{eff}(i), [\pi_{n,i}]_n, [(t_n(i,j))]_j, (L\alpha_{r,i}, L\xi_{r,i}, \mu_{r,i}, [Z_{n,r}(i)]_n, \nabla_i(r))_r \right)$$

(called auxiliary approximation of the sample  $i$ );

- $$P_{\xi}^{(k)}(m,i) = \left( R_{eff}(i), [\pi_{n,i}]_n, [(t_n(i,j))]_j, (L\xi_{r,i}, \mu_{r,i}, [Z_{n,r}(i)]_n, \nabla_i(r))_r \right)$$

(called particle approximation of the filtering distribution at  $i$ );

30

- $$P_{\omega}^{(k)}(m,i) = \left( R'_{eff}(i), [(t_n(i,j))]_j, (\omega_{r,i}, \nu_{r,i}, [Z'_{n,r}(i)]_n)_r \right) \quad \text{(called particle approximation of the smoothing distribution at } i \text{)}.$$

Sets of additional multiple particles are defined in the same format:

- 5     -      $P_{\alpha}^{(k)}(m, i) = \left( \dot{R}_{\text{eff}}(i), [\pi_{n,i}]_n, [(t_n(i, j))]_j, (L\dot{\alpha}_{r,i}, L\dot{\xi}_{r,i}, \dot{\mu}_{r,i}, [\dot{Z}_{n,r}(i)]_n, \dot{V}_i(r))_r \right)$   
produced by the module 2.1.1 of figure 4 and used by the module 2.1.2 (called re-sampled auxiliary approximation);
- $P_{\tilde{\omega}}^{(k)}(m, i) = \left( \tilde{R}_{\text{eff}}(i), [(t_n(i, j))]_j, (\tilde{\omega}_{r,i}, \tilde{v}_{r,i}, [\tilde{Z}_{n,r}(i)]_n)_r \right)$  produced  
10     by the module 2.1.4 of figure 4 and used by the module 2.2 (called mixed approximation of the smoothing distribution at  $i$ ).

These sets are data exchange units between the sub-modules that make up the particle EM algorithm implemented by the modules 2.1.1-2.1.4 and 2.2 of figure 4. Note that, in the detail of these sets, the frame indices  $\mathbf{y}_m$  and iteration indices  $k$  have been understood to lighten the notations. The notation  $P_{\omega}^{(k)}(m, 1:l)$  (respectively  $P_{\tilde{\omega}}^{(k)}(m, 1:l)$ ) is used to designate  
20     the set made up of  $P_{\omega}^{(k)}(m, 1), \dots, P_{\omega}^{(k)}(m, l)$  (respectively  $P_{\tilde{\omega}}^{(k)}(m, 1), \dots, P_{\tilde{\omega}}^{(k)}(m, l)$ ).

The sets  $P_{\alpha}(m, i)$  and  $P_{\xi}(m, i)$  also have the property of  
25     being classed according to lexicographical order over  $C_1^{2j+1} \times \dots \times C_n^{2j+1} \simeq (C_1 \times \dots \times C_n)^{2j+1}$ :  
for any  $r, \dot{r} \in [1, R_{\text{eff}}(i)], r \leq \dot{r} \Rightarrow [Z_{n,r}(i)]_n \preceq [Z_{n,\dot{r}}(i)]_n$ .

In the method presented, a calculation of the form  
30      $\log \sum_{i \in I} e^{b_i}, b_i < 0$  is performed many times. A primitive performance of this calculation raises cases of divergences. An appropriate digital procedure, detailed hereinbelow, is therefore introduced. A calculation of

the form  $\log \sum_{i \in I} e^{b_i}$  performed with this procedure is hereinbelow represented symbolically by  $\sum_{i \in I}^* b_i$ .

- If  $I = \emptyset$ ,  $\sum_{i \in I}^* b_i = 0$
- 5 - If  $I \neq \emptyset$ , let  $i^* = \operatorname{argmax}_{i \in I} b_i$

$$\sum_{i \in I}^* b_i = b_{i^*} + \log 1p \left( \exp \left\{ \sum_{i \in I \setminus i^*}^* (b_i - b_{i^*}) \right\} \right)$$

( $\log 1p(\cdot)$  is a standard function of digital computation  
10 libraries, accurately performing the evaluation of  
 $\log(1 + \epsilon)$  when  $|\epsilon| \ll 1$ ).

The particle estimation methods involve a transition  
kernel (function of the state space squared in the  
15 range  $[0, 1]$ ).

Since the wave functions are likely not to be causal,  
the transition kernels studied and used in the  
documents cited in the prior art are not therefore  
20 appropriate to the problem upon which we are focused  
here.

We take an integer  $\delta$  of the order of  $J \cdot \max T_n / T_e$ . For any  
 $i$ , there are defined

25

$$W_i([X_n(i)]_n, [X_n(i-1)]_n) =$$

$$\sum_{[X_n(i+1)]_n} \cdots \sum_{[X_n(i+\delta)]_n} \mathbf{P}_{\theta^{(k)}}(y_i, \cdots, y_{i+\delta}, [X_n(i-1)]_n, [X_n(i)]_n, \cdots, [X_n(i+\delta)]_n),$$

$$W_i([X_n(i-1)]_n) = \sum_{[X_n(i)]_n} W_i([X_n(i)]_n, [X_n(i-1)]_n)$$

and  $w_i([X_n(i)]_n, [X_n(i-1)]_n) = W_i([X_n(i)]_n, [X_n(i-1)]_n) / W_i([X_n(i-1)]_n)$   
which is a transition kernel of  $C_1^{2J+1} \times \cdots \times C_n^{2J+1}$  in  
30 itself.

Approximations of Monte-Carlo type are used in practice to evaluate this kernel.

5 The module 2, and in particular the module 2.1 performing the step E, are broken down into sub-modules as detailed in figure 4.

- for each iteration  $k$ 
  - o for each sample  $i$ 
    - 10 o the module 2.1.1 corresponds to a step of re-sampling of the particles which takes as input the set of particles  $P_{\alpha}^{(k)}(m, i-1)$  and gives as output the set of re-sampled particles  $P_{\alpha}^{(k)}(m, i-1)$ ;
    - 15 o the module 2.1.2 corresponds to a step of particle filtering which takes as input  $P_{\alpha}^{(k)}(m, i)$ , the frame  $y_m$ , the estimation of the parameter vector  $\theta_m^{(k)}$  obtained on the preceding iteration and which gives as output 20 the filtering distribution  $P_{\xi}^{(k)}(m, i)$  for the sample  $i$  as well as the auxiliary approximation  $P_{\alpha}^{(k)}(m, i)$  for the samples  $i$ ;
    - 25 o the module 2.1.3 corresponds to a particle smoothing step. In one embodiment, this particle smoothing step implements a so-called "fixed-lag smoothing" algorithm which takes as input the filtering distribution  $P_{\xi}^{(k)}(m, i)$  of the sample  $i$  and gives as output 30 the smoothing distribution  $P_{\omega}^{(k)}(m, i-\Delta)$  of the sample  $i-\Delta$ ,  $\Delta$  being an integer parameter defined on the instantiation of the system;
    - 35 o the smoothing distributions  $P_{\omega}^{(k)}(m, i-\Delta)$  for  $i=1, \dots, I$  are aggregated in a buffer (see figure 4) in order to produce as output the

- smoothing distribution of the entire frame
- $$P_{\omega}^{(k)}(m, 1: I);$$
- o the module 2.1.4 corresponds to an optional, so-called stochastic mixing step which
- 5 transforms  $P_{\omega}^{(k)}(m, 1: I)$  into  $P_{\tilde{\omega}}^{(k)}(m, 1: I)$  (possibly  $P_{\tilde{\omega}}^{(k)}(m, 1: I) = P_{\omega}^{(k)}(m, 1: I)$ );
- o the module 2.2 performs the step M from the frame  $\mathcal{Y}_m$  and the particles  $P_{\tilde{\omega}}^{(k)}(m, 1: I)$  and supplies, after K iterations, the final
- 10 output of the parameter vector  $\theta_m^{(K)}$  of the module 2 for the frame  $\mathcal{Y}_m$ .

In the descriptions of the submodules 2.1.1 to 2.1.4, the frame indices  $m$  and iteration indices  $k$  are

15 understood.

The algorithm implemented in the re-sampling module 2.1.1 comprises:

20 - as input for the module 2.1.1:

- o  $P_{\alpha}^{(k)}(m, i-1) = \left( R_{\text{eff}}(i-1), [\pi_{n,i-1}]_n, [(t_n(i-1), j)]_n, (L\alpha_{r,i-1}, L\xi_{r,i-1}, \mu_{r,i-1}, [Z_{n,r}(i-1)]_n, \nabla_{i-1}(r))_r \right)$
- for the variable  $(Y_1, \dots, Y_{R_{\text{eff}}(i-1)}) \sim \text{Mult}_R(\alpha_{1,i-1} \cdot \mu_{1,i-1}, \dots, \alpha_{R_{\text{eff}}(i-1),i-1} \cdot \mu_{R_{\text{eff}}(i-1),i-1})$
- 25 - assume  $\dot{R}_{\text{eff}}(i-1) = \text{card} \{r \in \llbracket 1, R_{\text{eff}}(i-1) \rrbracket, Y_r > 0\}$
- define:  $\epsilon_0: \llbracket 1, \dot{R}_{\text{eff}}(i-1) \rrbracket \rightarrow \llbracket 1, R_{\text{eff}}(i-1) \rrbracket$  by  $\epsilon_0(\dot{r}) = \inf \{r, \sum_{r' \leq r} \min(Y_{r'}, 1) \geq \dot{r}\}$
- for any  $1 \leq r \leq \dot{R}_{\text{eff}}(i-1)$ , assume:
  - o for any  $1 \leq n \leq N$ ,
  - 30 o  $\dot{Z}_{n,r}(i-1) = Z_{n,\epsilon_0(r)}(i-1)$
  - o  $L\dot{\alpha}_{r,i-1} = \log 1/R$
  - o  $\dot{\mu}_{r,i-1} = Y_{\epsilon_0(r)}$
  - o  $L\dot{\xi}_{r,i-1} = L\xi_{\epsilon_0(r),i-1} - L\alpha_{\epsilon_0(r),i-1}$

- $\dot{\mathbf{V}}_{i-1}(r) = \epsilon_0(r)$

- as output for the module 2.1.1:

$$P_{\alpha}^{(k)}(m, i-1) =$$

$$\left( \dot{R}_{eff}(i-1), [\pi_{n,i-1}]_n, [(t_n(i-1, j))]_j \right)_n,$$

$$\left( L\dot{\alpha}_{r,i-1}, L\dot{\Sigma}_{r,i-1}, \dot{\mu}_{r,i-1}, [\dot{Z}_{n,r}(i-1)]_n, \dot{\mathbf{V}}_{i-1}(r) \right)_r$$

○

5

The algorithm implemented by the new generation sampling module 2.1.2 comprises:

- as input for the module 2.1.2 of the sampling module:

10

- $P_{\alpha}^{(k)}(m, i-1) =$

$$\left( \dot{R}_{eff}(i-1), [\pi_{n,i-1}]_n, [(t_n(i-1, j))]_j \right)_n, \left( L\dot{\alpha}_{r,i-1}, L\dot{\Sigma}_{r,i-1}, \dot{\mu}_{r,i-1}, [\dot{Z}_{n,r}(i-1)]_n, \dot{\mathbf{V}}_{i-1}(r) \right)_r$$

- the frame  $\mathbf{y}_m$

- the parameter vector  $\theta_m^{(k-1)}$

15

- updating of the time differences and of the jump indicators:

- for any channel  $n$  and any index  $j$ ,

$$\tilde{t}_n(i, j) = t_n(i-1, j) - T_e$$

20

- if  $\tilde{t}_n(i, j) \leq 0$ , for any  $j$ ,  $t_n(i, j) = \tilde{t}_n(i, j) + T_n^{(k)}$  and  $\pi_{n,i} = 1$

- otherwise, for any  $j$ ,  $t_n(i, j) = \tilde{t}_n(i, j)$  and  $\pi_{n,i} = 0$

- assume  $\kappa = \{\kappa_1, \dots, \kappa_K\} = \{1 \leq n \leq N, \pi_{n,i} = 1\}$  and

$$\bar{\kappa} = \{\bar{\kappa}_1, \dots, \bar{\kappa}_K\} = \{1 \leq n \leq N, \pi_{n,i} = 0\}.$$

25

- for any  $r \leq \dot{R}_{eff}(i-1)$ ,

- for any  $d(\kappa_1, \dots, \kappa_K) = (d_{\kappa_1}, \dots, d_{\kappa_K}) \in C_{\kappa_1} \times \dots \times C_{\kappa_K}$

- for any  $n \in \llbracket 1, N \rrbracket$ ,

- define  $Z_{n,r,d(\kappa_1, \dots, \kappa_K)}(i) = \dot{Z}_{n,r}(i-1)$  if  $n \notin \kappa$

- $Z_{n,r,d(\kappa_1,\dots,\kappa_K)}(i) = [d_{\kappa_{\blacksquare}}, \dot{z}_{n,r,2}(i-1), \dots, \dot{z}_{n,r,2l}(i-1)]_{\text{if}}$
- $n = \kappa_{\blacksquare} \in \kappa$
- calculate  $LW_i([Z_{n,r,d(\kappa_1,\dots,\kappa_K)}(i)]_n | \dot{Z}_{n,r}(i-1))$
- calculate
- 5  $LW_i([Z_{n,r}(i-1)]_n) = \sum_{d(\kappa_1,\dots,\kappa_K)}^* LW_i([Z_{n,r,d(\kappa_1,\dots,\kappa_K)}(i)]_n | \dot{Z}_{n,r}(i-1))$
- normalize, for any
- $d(\kappa_1, \dots, \kappa_K), LW_i([Z_{n,r,d(\kappa_1,\dots,\kappa_K)}(i)]_n | \dot{Z}_{n,r}(i-1)) =$
- $LW_i([Z_{n,r,d(\kappa_1,\dots,\kappa_K)}(i)]_n | \dot{Z}_{n,r}(i-1)) - LW_i([Z_{n,r}(i-1)]_n)$
- draw the variable
- 10  $(\rho_{d(\kappa_1,\dots,\kappa_K)}) \sim \text{Mult}_{\dot{\mu}_{r,i-1}}(w_i([Z_{n,r,d(\kappa_1,\dots,\kappa_K)}(i)]_n | \dot{Z}_{n,r}(i-1)))$
- assume for any
- $r, R_{eff}(r, i) = \text{card} \{d(\kappa_1, \dots, \kappa_K), \rho_{d(\kappa_1,\dots,\kappa_K)} > 0\}$
- define the bijection
- $\epsilon_{1,r}: \llbracket 1, R_{eff}(r, i) \rrbracket \rightarrow \kappa, \epsilon_{1,r}(\dot{r}) = \inf\{d(\kappa'), \kappa' \in \kappa, \sum_{d(\kappa'') \leq d(\kappa')} \rho_{d(\kappa'')} \geq \dot{r}\}$
- 15 - note, for any n, any  $\dot{r} \in \llbracket 1, R_{eff}(r, i) \rrbracket$
- o  $Z_{n,(r,\dot{r})}(i) = Z_{n,r,\epsilon_{1,r}(\dot{r})}(i)$
- o  $LA_{n,(r,\dot{r}),i} = LW_i([Z_{n,r}(i-1)]_n)$
- o  $\mu_{n,(r,\dot{r}),i} = \rho_{\epsilon_{1,r}(\dot{r})}$
- o  $L\Xi_{n,(r,\dot{r}),i} = L\dot{\Xi}_{n,r,i-1} + \log P_{\theta^{(k)}}(y(i) | [Z_{n,(r,\dot{r}),i}]_n) - LA_{n,(r,\dot{r}),i}$
- 20 o  $\nabla_i(r, \dot{r}) = \dot{\nabla}(r)$
- assume  $R_{eff}(i) = \sum_r \dot{R}_{eff}(i-1) R_{eff}(r, i)$
- define the bijection
- 25  $(0_1, 0_2): \llbracket 1, R_{eff}(i) \rrbracket \rightarrow \bigcup_{r=1}^{r \leq \dot{R}_{eff}(i-1)} \{r\} \times \llbracket 1, R_{eff}(r, i) \rrbracket$  a
- permutation such that
- $r_1 \leq r_2 \Rightarrow [Z_{n,(0_1(r_1), 0_2(r_1))}(i)]_n \leq [Z_{n,(0_1(r_2), 0_2(r_2))}(i)]_n$
- for any  $r \leq R_{eff}(i)$ ,
- 30 o  $[Z_{n,r}(i)]_n = [Z_{n,(0_1(r), 0_2(r))}(i)]_n$
- o  $LA_{r,i} = LA_{(0_1(r), 0_2(r)),i}$
- o  $\mu_{r,i} = \mu_{(0_1(r), 0_2(r)),i}$

- $L\Xi_{r,i} = L\Xi_{(O_1(r), O_2(r)),i}$
- $\nabla_i(r) = \nabla_i(O_1(r), O_2(r))$

- normalize the weights, for any  $r$ ,

5

- $LA_i = \sum_r^* LA_{r,i}$
- $L\Xi_i = \sum_r^* L\Xi_{r,i}$
- $L\alpha_{r,i} = LA_{r,i} - LA_i$
- $L\xi_{r,i} = L\Xi_{r,i} - L\Xi_i$

10

- at the output:

$P_\xi^{(k)}(m, i) = \left( R_{\text{eff}}(i), [\pi_{n,i}]_n, \left( L\xi_{r,i}, \mu_{r,i}, [Z_{n,r}(i)]_n, \nabla_i(r) \right)_r \right)$  is the filtering distribution of the sample  $i$ ;

$P_\alpha^{(k)}(m, i) = \left( R_{\text{eff}}(i), [\pi_{n,i}]_n, \left( L\alpha_{r,i}, L\xi_{r,i}, \mu_{r,i}, [Z_{n,r}(i)]_n, \nabla_i(r) \right)_r \right)$  is the auxiliary distribution of the sample  $i$ .

15

For the smoothing step implemented by the module 2.1.3 (figure 4), the method known by the name "fixed-lag smoothing" is for example used. We take an integer  $\Delta$ ,

20

- at the input, the filtering distribution of the sample  $i$ :

$$P_\xi(m, i) = \left( R_{\text{eff}}(i), [\pi_{n,i}]_n, \left( L\xi_{r,i}, \mu_{r,i}, [Z_{n,r}(i)]_n, \nabla_i(r) \right)_r \right)$$

- assume  $\nabla_{i,0}(r) = \nabla_i(r)$

- for any  $1 \leq j \leq \Delta$

25

$$\circ \quad \nabla_{i,j}(r) = \nabla_{i-j}(\nabla_{i,j-1}(r))$$

- assume  $R'_{\text{eff}}(i - \Delta) = R_{\text{eff}}(i)$

- assume, for any  $r \leq R_{\text{eff}}(i)$ ,

30

- $v_{r,i-\Delta} = \mu_{r,i}$
- $\omega_{r,i-\Delta} = \xi_{r,i}$
- $[Z'_{n,r}(i - \Delta)]_n = [Z_{n,\nabla_{i,\Delta}(r)}(i)]_n$

- at the output, the set

$P_{\omega}(m, i - \Delta) = \left( R'_{\text{eff}}(i - \Delta), \left( \omega_{r, i - \Delta}, v_{r, i - \Delta}, [Z'_{n, r}(i - \Delta)]_n \right)_r \right)$  is a particle approximation of the smoothing distribution at  $i - \Delta$ .

5

The module 2.1.4 is configured to implement a step of stochastic mixing of the particles obtained by the module 2.1.3. It therefore takes as input the set of the smoothed particles  $P_{\omega}(m, 1: I)$  and returns a new set of particles denoted  $P_{\tilde{\omega}}(m, 1: I)$ . This module 2.1.4 makes it possible to safeguard against the risks of convergences toward a local optimum of the EM algorithm. It consists, for any  $i$ , in retaining from  $P_{\omega}(m, i)$  only a single particle drawn at random. It should be noted that, if this step is applied on each iteration  $k$  of each frame  $y_m$ , the algorithm becomes unstable. Therefore, it is applied only to certain pairs  $(m, k)$ , those which belong to a set  $\mathcal{I}_t$ , this set being to be increasingly less dense with the evolution of  $(m, k)$ . For example, it is possible to consider  $\mathcal{I}_t$  such that,  $\mathcal{I}_t$  ssi  $(m, k) \in (m - 1) \cdot K + k$  is the square of an integer. The procedure implemented in this module is:

- as input:

$P_{\omega}^{(k)}(m, 1: I) = \left( \left( R'_{\text{eff}}(i), \left[ (t_n(i, j))_j \right]_n, \left( \omega_{r, i}, v_{r, i}, [Z'_{n, r}(i)]_n \right)_r \right) \right)_{i=1 \dots I}$  is the approximation of the smoothing distribution;

- if  $(m, k) \in \mathcal{I}_t$

- o for any  $i \in \llbracket 1, I \rrbracket$
- o assume  $\check{R}_{\text{eff}}(i) = 1$
- o draw an index  $r_0$  in  $\llbracket 1, R'_{\text{eff}}(i) \rrbracket$  with a probability proportional to  $\omega_{i, r_0} \cdot v_{i, r_0}$
- o assume  $\check{\omega}_{1, i} = 1, \check{v}_{1, i} = 1$  and  $[\check{Z}_{n, 1}(i)]_n = [Z'_{n, r_0}(i)]_n$

- if  $(m, k) \notin \mathbf{It}$ 
    - o for any  $i \in \llbracket 1, I \rrbracket$ ,
    - o assume  $\check{R}_{eff}(i) = R'_{eff}(i)$
    - 5 o for any  $r \in \llbracket 1, \check{R}_{eff}(i) \rrbracket$
    - o assume  $\check{\omega}_{r,i} = 1, \check{\nu}_{r,i} = 1$  and  $[\check{Z}_{n,r}(i)]_n = [Z'_{n,r}(i)]_n$
  - as output:

$$P_{\check{\omega}}(m, 1:I) = \left( \left( \check{R}_{eff}(i), \left[ (t_n(i, j))_j \right]_n, \left( \check{\omega}_{r,i}, \check{\nu}_{r,i}, [\check{Z}_{n,r}(i)]_n \right)_r \right) \right)_{i=1 \dots I}$$
is
- 10 the mixed approximation of the smoothing distribution.

The module 2.2, shown in figures 3 and 4, corresponds to the step M of the EM algorithm.

15 On iteration  $k$ , it takes as input the signal frame  $y_m$ , the parameter vector  $\theta_m^{(k-1)}$  estimated on the preceding iteration and the mixed particles  $P_{\check{\omega}}(m, 1:I)$  obtained at the end of the step E (that is to say by the module 2.1 and in particular the submodule 2.1.4). It gives as

20 output a refined estimation  $\theta_m^{(k)}$  of the parameter vector.

This estimation is chosen to make the conditional expectation of the log-likelihood of the observations decrease according to the parameter vector  $\theta$ :

25 
$$L_{\theta_m^{(k-1)}}(\theta_m^{(k)}) < L_{\theta_m^{(k-1)}}(\theta_m^{(k-1)}).$$

Many methods can be envisaged for this purpose. In one embodiment, the following procedure is put in place:

30

The parameter vector  $\theta_m^{(k)}$  estimated at the  $k$ th iteration on the  $m$ th frame  $\theta_m^{(k)} = ([A_{nm}^{(k)}, T_{nm}^{(k)}, \delta f_{nm}^{(k)}, \tau_{nm}^{(k)}]_{n=1 \dots N}, \sigma_m^2)^{(k)}$  is

denoted in the form  $\theta_m^{(k)} = (\theta_{m,l}^{(k)})_{l=1,\dots,L}$  where  $L$  is the total number of parameters to be estimated. The permutation between the parameters and the indices  $l$  can be any permutation, even random, and different on each iteration.

- note  $\theta_m^{(k,0)} = \theta_m^{(k-1)}$
- for any  $l, 1 \leq l \leq L$
- the  $i$ th parameter is updated according to

10

$$\theta_l^{(k)} = \theta_l^{(k-1)} - \lambda \cdot \frac{\sum_i \sum_r \check{\omega}_{r,i} \check{\nu}_{r,i} \left. \frac{\partial L'_{\theta}}{\partial \theta_l} \right|_{\theta=\theta^{(k,l-1)}} (y(i) | [\check{Z}_{n,r,i}]_n)}{\sum_i \sum_r \check{\omega}_{r,i} \check{\nu}_{r,i} \left. \frac{\partial^2 L'_{\theta}}{\partial \theta_l^2} \right|_{\theta=\theta^{(k,l-1)}} (y(i) | [\check{Z}_{n,r,i}]_n)}$$

- the following is assumed:  
 $\theta^{(k,l)} = (\theta_1^{(k)}, \dots, \theta_{l-1}^{(k)}, \theta_l^{(k-1)}, \theta_{l+1}^{(k-1)}, \dots, \theta_L^{(k-1)})$  and
- 15 -  $\theta_m^{(k)} = \theta_m^{(k,L)}$  is obtained as output.

The 1<sup>st</sup> and 2<sup>nd</sup> order derivatives are calculated analytically. The parameter  $\lambda \in ]0,1]$  makes it possible to adjust the speed of convergence of the algorithm. It is possibly different for each parameter.

Possibly, if some parameters of the system are known a priori with certainty, they can advantageously be considered as constants in the algorithm and are not updated in the step M. In this particular case, the convergence of the algorithm is accelerated and the residual variance due to the estimation of these parameters is neutralized.

30 Figure 5 details the interactions between the module 3 for the long-term tracking of the parameter vector  $\theta$  of the composite signal  $y$  and the demodulation module 4.

The module 3 for the long-term tracking of the parameter vector  $\theta$  forms an estimation  $\theta_m$  of the parameter vector  $\theta$  by linearly combining all the estimations obtained for the parameter vectors  $\theta_1^{(K)}, \theta_2^{(K)}, \dots, \theta_m^{(K)}$  by the EM algorithm (implemented by the module 2) over the frames  $y_1$  to  $y_m$ .

This estimation can notably be done recursively from:

- 10 -  $\theta_m^{(K)}$  the parameter vector estimated for the frame  $y_m$  over  $K$  iterations by the module 2 and in particular by the module 2.2; and from
- $\theta_{m-1}$  the parameter vector estimated in the long term by the module 3 for the long-term tracking of the parameter vector  $\theta$  over the preceding frame if  $m > 1$  or possibly by the priming module 1.3 if  $m = 1$ .

At the convergence of the EM algorithm on the frame  $y_m$ , all the parameter estimators have a variance proportional to  $\sigma_m^{2(K)}$ .

The following is therefore assumed:

$$\tilde{\gamma}_m = \frac{\gamma \cdot \sigma_m^{2(K)}}{\gamma \cdot \sigma_m^{2(K)} + \tilde{\sigma}_{m-1}^2}$$

$$\tilde{\sigma}_m^2 = \frac{1}{\frac{\gamma}{\tilde{\sigma}_{m-1}^2} + \frac{1}{\sigma_m^{2(K)}}}, \quad \tilde{\sigma}_0^2 = +\infty$$

25

Then the parameters are updated according to:

$$\theta_m = \theta_{m-1} + (1 - \tilde{\gamma}_m) \cdot (\theta_m^{(K)} - \theta_{m-1})$$

30 The tracking parameter  $\gamma$  set by the user between 0 and 1 makes it possible to give the elasticity to take

account of the evolution of the channel. The quantities  $\tilde{\gamma}_m$  and  $\tilde{\sigma}_m^2$  are respectively a tracking parameter and a variance that are modified in order to linearly and optimally combine the estimation on the frame  $m$  of the EM algorithm  $\theta_m^{(k)}$  and the long-term estimation up to the frame  $m-1, \theta_{m-1}$ .

The demodulation module 4, detailed in figure 5, demodulates the channels. This module 4 implements a stochastic Viterbi algorithm. This demodulation module 4 essentially produces the symbol frames estimated  $\hat{s}_{1,m}, \hat{s}_{2,m}, \dots, \hat{s}_{N,m}$  for each of the channels.

The demodulation module 4 comprises two functional blocks.

The submodule 4.1 is an instantiation of the step E of the EM algorithm, limited to the reproduction of the modules 2.1.1 and 2.1.2 corresponding to the particle filtering step. This submodule 4.1 takes as input:

- the frame  $y_m$ ;
- the parameter vector  $\theta_m$  estimated in the long term on the frame  $y_m$  by the module 3 for the long-term tracking of the parameter vector  $\theta$ .

The submodule 4.1 performs, with this parameter vector  $\theta_m$ , a new filtering step and produces as output:

- a set of particles  $\hat{P}(m, 1: I)$  intended for the submodule 4.2. This set can, depending on the particle approximation used, be a set of particles or a set of multiple particles. It should be noted, that in this output, the filtering weights obtained are not used thereafter;
- an auxiliary particle approximation  $P_a(m, I)$  of the last sample of the frame  $y_m$ . It is this

approximation which is used to initialize the module 2 and in particular the module 2.1.1 of the frame  $\mathcal{Y}_{m+1}$  via their input parameter  $P_\alpha(m+1,0)$ .

- 5 The submodule 4.2 performs the demodulation of the channels by a stochastic Viterbi algorithm. It takes as input:
- the parameter vector  $\theta_m$  estimated in the long term up to the frame  $\mathcal{Y}_m$  by the module 3 for the long-term tracking of the parameter vector  $\theta$ ;
  - the support of the particle approximation  $\hat{P}(m,1:I)$  of the frame given by the submodule 4.1.
- 15 Through the stochastic Viterbi algorithm, the a posteriori maximum vectorized sequence  $([\hat{s}_n(i)]_n)$  is obtained. The symbols transmitted over each channel are estimated by the sequence  $(\hat{s}_n(i_n))_{1 \leq i_n \leq I_n}$  according to the procedure:
- as input: the a posteriori maximum vector sequences:  $([\hat{s}_{n,i}]_n)$  and the jump indicators  $[\pi_{n,i}]$  for any  $1 \leq i \leq I$  and any  $1 \leq n \leq N$ ;
  - for any  $n \in \llbracket 1, N \rrbracket$ 
    - o assume  $I_n = \sum_{i=1}^I \pi_{n,i}$
    - o for any  $i_n \in \llbracket 1, I_n \rrbracket$
    - o  $\hat{e}(i_n) = \inf\{i \in \llbracket 1, I \rrbracket, \sum_{i'=1}^i \pi_{n,i'} \geq i_n\}$
    - o  $\hat{s}_n(i_n) = \hat{s}_{n,1}(\hat{e}(i_n))$
  - as output, the sequences  $\hat{s}_{n,m} = (\hat{s}_n(i))_{i \in \llbracket 1, I_n \rrbracket}$  are an estimation of the symbols transmitted by the channel  $n$  on the frame  $\mathcal{Y}_m$ .

The preprocessing module 1 (see figure 2) ensures, in particular, the formation of the frames  $\mathcal{Y}_m$ .

This preprocessing module 1 takes as input the information stream  $y$ . In the case of the identification of Double Talk signals for example, the signals of this composite signal  $y$  are not transmitted according to raised cosine filters  $h$  but according to a raised route cosine filter  $\tilde{h}$ . In this case, the signal directly observed and modulated by raised route cosine filters is denoted  $\tilde{y}$  and the filtered signal which is manipulated by the downstream modules is denoted  $y$ . It therefore falls to the preprocessing module 1 to filter the signal picked up by  $\tilde{h}$  to obtain a signal formed by  $h$ .

In addition, at the moment of the interception, the symbol times are unknown, the signal has to be greatly upsampled in order to be sure of observing the Shannon condition. Once the symbol times are known, there is no longer a need to manipulate a greatly upsampled signal.

As shown in figure 6, the module 1 comprises three submodules.

- the submodule 1.1 ensures the "buffering" of the signal  $y$  under observation at its input to form the frames  $\tilde{y}_m$ ;
- the submodule 1.2 proceeds jointly with the downsampling and the filtering of the frame  $\tilde{y}_m$  to give, as output, the frame  $y_m$  which is the signal directly manipulated by the other modules of the blind separation system 10. This submodule 1.2 uses the more or less accurate knowledge of the symbol times  $T_n$  available in the system through the vector  $\theta_{m-1}$  delivered by the module 3 for the long-term tracking of the parameter vector  $\theta$ . For the first frame, the parameter vector  $\theta_0$  is estimated for example blind by the submodule 1.3;

- the submodule 1.3 makes it possible to perform a rough first estimation of the parameter vector  $\theta$ . This step is used only on the first frame of the signal and is called priming step. For example, the frequencies can be primed from the property of circularity of the constellations considered, the symbol times by the spectral line method, the amplifications by an equitable sharing of the energy of the signal intercepted, the delay times and the phase errors which are a priori bounded can be taken equal to 0 or to values drawn randomly.

The procedure of joint filtering and decimation of the signal implemented in the adaptation submodule 1.2 is as follows:

- as input: the frame  $\tilde{y}_m(i) = \sum_{n=1}^N A_n \cdot e^{i2\pi\delta f_N \left( \sum_{j=-\infty}^{j=+\infty} \tilde{h}_n(iT_{e0} - jT_n - \tau_n) s_n(j) \right)} + \eta(i)$  sampled at the rate  $T_{e0}$ ;  $\tilde{h}_n$  is a raised cosine filter of symbol time  $T_n$ ; the  $T_n$  are substantially equal;
- an estimation of the sampling times of each channel  $T_1, \dots, T_N$  via the parameter  $\theta_{m-1}$ ;
- assume  $T_e = \frac{1}{2} \min T_n$  and  $\tilde{h}$  the raised cosine filter of symbol time  $2T_e$ ;
- $\tilde{y}_m$  is filtered and downsampled jointly to  $y_m$  at the rate  $T_e$  according to the equation

$$y_m(i) = \frac{T_e}{T_{e0}} \sum_j \tilde{y}_m(j) \cdot \tilde{h}(jT_{e0} - iT_e)$$

- as output, the frame  $y_m$  is given as input for the downstream processing chain.

The proposed separation algorithm is a generalization of the EM algorithm calculated on a particle

approximation. This algorithm is applied to an analytical model of the composite signal to extract from it the source signals. The decoding is ensured by a stochastic Viterbi algorithm.

5

The term real time is justified here in as much as the signal is "buffered" in frames of  $I$  (100 to a few thousand) symbols and in that the temporal complexity of the separation algorithm and its memory resource requirements are linear according to the size of the frame. Thus, it is possible to implement it in such a way that the processing time for a frame is equal to the transmission time for its  $I$  samples whatever the size value.

15

Thus, the size of the frames can be adapted to signals whose parameters are constant in time (long frames) or to signals whose parameters vary rapidly in time (short frames), the long-time tracking module is in this last case responsible for composing an estimation of the parameters of lesser variance.

Advantageously, the systems and methods described above introduce the notion of particles with multiplicity which makes it possible to greatly limit the number of particles to be processed and therefore significantly speed up the execution of the algorithm. In effect, with this paradigm, the effective number of particles used for a frame in an EM iteration adapts automatically to the difficulty of this processing. Thus, on the first iterations of the first frames, many particles are effectively used; when the convergence of the parameters has begun, this number decreases gradually before stabilizing at a number of particles much smaller than that which would be necessary in the conventional particle filtering paradigm when the convergence is acquired.

Advantageously, the methods and systems described above make it possible to process a wide range of configurations:

- 5    - different types of modulations (for example M-ASK, M-PSK, Q-PSK, M-QAM), the type of modulation can be different for each of the channels;
- varied frequency bands for the different channels, whether they are different or equal (or substantially  
10 equal because of synchronization errors between the channels to be envisaged for example for the separation of Double Talk signals);
- varied symbol bit rates for the different channels, whether they are different or equal (or substantially  
15 equal because of synchronization errors between the channels, to be envisaged for example for the separation of Double Talk signals);
- transmission gains at the different sources, whether they are different or substantially equal;
- 20    - there can be any delay times and phase errors on each channel.

The systems described are capable of automatically detecting, optimally combining and taking advantage of  
25 any asymmetry between the source signals.

In an illustrative implementation of the blind separation method described above, a composite signal, comprising two "DoubleTalk® Carrier-in-Carrier®"  
30 signals is simulated according to the following parameters:

- the signal  $y$  is made up of two channels:  $N = 2$ ;
- the sampling period on interception is taken as  
35 reference time unit:  $T_e = 1$ ;
- the forming filters  $\tilde{h}_1$  and  $\tilde{h}_2$  are raised cosine filters of 0.35 roll-off;

- the symbol times are  $T_1 = 2$  and  $T_2 = 2$  in reduced units;
- the carrier residues are  $\delta f_1 = 1.10^{-5}$ ,  $\delta f_2 = 3.10^{-6}$  in reduced units;
- 5 - the phase shifts are  $\varphi_1 = \pi/10$  and  $\varphi_2 = \pi/4$ ;
- the delays are  $\tau_1 = -0.3 \cdot T_1$  and  $\tau_2 = -0.8 \cdot T_2$ ;
- the two modulations are QPSK;
- the amplifications are such that  $|A_2| = 0.95 \cdot |A_1|$
- the Gaussian noise is white, the ratio between the  
10 average power of the signals and that of the noise  
is 20 dB.

The composite signal simulated is analyzed over 20 frames of a size equivalent to 500 symbols per channel.  
15 After a signal adaptation step, the blind separation method is instantiated with the following settings:

- the maximum number of particles is set at 200;
- the smoothing parameter is  $\Delta = 40$ ;
- 20 - the channel is assumed quasi-invariant:  $\gamma = 0.9$ ;
- there are  $K = 10$  EM iterations per frame.

The progress of the separation algorithm on this composite signal is shown in figures 7 to 14 and in  
25 figure 15.

Figures 7 to 14 visually show the results of a few typical iterations of the algorithm, the sub-figures respectively represent, from left to right and from top  
30 to bottom:

- a reconstruction of the channel 1 in the complex plane (this reconstruction is obtained by  
35 subtracting from the composite signal the estimation of the channel 2 and the inter-symbol interferences on the channel 1. The grey levels represent the local density of the measurements on

- the reconstructed channel 1). The SINR value is an estimation in decibels of the ratio between the power of the signal on the channel 1 and all the disturbances which are overlaid thereon (inter-symbol interferences, channel 2 and noise).
- 5       - a reconstruction of the channel 2 (this reconstruction is obtained by subtracting from the composite signal the estimation of the channel 1 and the inter-symbol interferences on the channel
- 10       2. The grey levels represent the local density of the measurements on the reconstructed channel 2). The SINR value is an estimation in decibels of the ratio between the power of the signal on the channel 2 and all the disturbances which are
- 15       overlaid thereon (inter-symbol interferences, channel 1 and noise).
- a reconstruction of the composite signal (this reconstruction is obtained by subtracting from the composite signal the inter-symbol interferences on
- 20       the channel 1 and on the channel 2. The grey levels represent a local density of measurements on the reconstructed composite signal. The black dots correspond to what would be the composite signal without noise, the pairs of digits above each dot
- 25       designate the pairs of states of the constellations of the channels 1 and 2 which correspond to this composite signal state). The quantitative data in this sub-figure are: Tr: the number of the frame processed, It: current iteration, MP: the average
- 30       effective number of particles used to process the current iteration and SI: standard deviation of the noise estimated on this iteration.
- a reconstruction in the frequency domain of the channel 1, of the channel 2 and of the noise.
- 35       - a reconstruction in the time domain of the forming filters on each of the channels.

Figure 7 shows the signals after the adaptation phase (0 iterations). They are indiscernible. The SINRs for the channels 1 and 2 cannot yet be estimated; their unknown value is denoted "?".

5

Figure 8 shows the signals after reconstruction with the initialization parameters of the algorithm (that is to say, after a single iteration): the noise level is high and relevant structures cannot be made out by the eye on the different reconstructions.

10

In figure 9, after four iterations, it can be seen that the noise level has decreased, a QPSK begins to be provided on the reconstruction of the channel 2.

15

Figure 10 shows the last iteration of the algorithm on the first frame; on each of the channels, a QPSK can be seen to form. The noise level is still high however, the algorithm has not finished converging at the end of the iterations allocated to the first frame.

20

However, it can be seen in figure 11 that the convergence continues on the second frame: the change of frame has not created any upset in the algorithm, the convergence mode is continuous.

25

It is on the fourth iteration of the second frame (figure 12) that the convergence of the algorithm is acquired. The QPSKs are well defined on each of the reconstructed channels and the estimated noise level is lower than the noise level of the intercepted signal (the adaptation step has also helped to save a few dB of SNR).

30

Figures 13 and 14 are typical of the rest of the results of the algorithm: the convergence of the parameters remains stable over time, the algorithm follows the evolution of the channel and corrects the

35

accumulation of the carrier residue estimation errors by making the phase errors evolve slightly in the course of the frames.

5 It should be noted that the paradigm of the particles with multiplicity is shown, in this example, to be effective for adapting the number of effective particles to the difficulty of the case to be processed. If in the first iterations 88 to 14  
10 effective particles are used (of the 200 available), this number stabilizes at 4-5 in the subsequent iterations, which is a significant gain in terms of resource time and hardware requirements compared to the prior art in blind demodulation.

15

Figure 15 presents the evolution of the estimation of the parameters in the course of the frames ("global" estimation) and of the iterations ("local" estimation). It will be noted in particular therein that the  
20 estimation of the phase errors changes slightly to compensate the variance on the estimation of the carrier residues.

## CLAIMS

1. A real-time method for blind separation and  
5 demodulation of digital telecommunications signals,  
called channels, from the observation, by means of a  
single sensor, of a composite signal comprising these  
signals, the parameters of these channels including  
10 their modulation type, their amplification, their  
phase-shift, their delay time at the sensor, their  
frequency and their symbol time, these parameters for  
the different channels being able to be different,  
substantially or perfectly equal, this method  
comprising the following steps:
- 15 - acquisition of a first plurality of observations  
of the composite signal performed by means of the  
signal sensor;
  - estimation, from the first plurality of  
observations acquired, of the parameters of the  
20 channels in terms of the "Maximum Likelihood" by  
an Expectation-Maximization algorithm, the  
conditional expectation of the log-likelihood  
being calculated, in this algorithm, recursively  
by a particle filtering-smoothing method;
  - 25 - joint demodulation of the channels according to a  
stochastic Viterbi algorithm.
2. The method of the preceding claim, in which the  
particle filtering-smoothing method comprises a  
30 particle approximation of the filtering distribution,  
this particle approximation assigning, in addition to  
weights, multiplicities to the particles of this  
approximation, a multiplicity being the number of  
particles representing a same state.
- 35
3. The method of claim 1 or 2, in which the particle  
filtering-smoothing method comprises a particle  
approximation of the smoothing distribution, this

particle approximation assigning, in addition to weights, multiplicities to the particles of this approximation, a multiplicity being the number of particles representing a same state.

5

4. The method of any one of the preceding claims, further comprising,

- a step of acquisition of a second plurality of observations of the composite signal performed by means of the single sensor;
- a step of estimation, from the second plurality of observations acquired, of the parameters of the channels in terms of the Maximum Likelihood by an Expectation-Maximization algorithm, the conditional expectation of the log-likelihood being calculated, in this algorithm, recursively by a particle filtering-smoothing method;
- a step of long-term estimation of the parameters, this long-term estimation of the parameters being associated with the second plurality of observations and being a linear combination of the parameters estimated from the first plurality of observations and of the estimated parameters of the second plurality of observations.

25

5. A computer program product implemented on a memory medium, capable of being implemented in a computer processing unit and comprising instructions for the implementation of a method as claimed in any one of the preceding claims.

30

Key to figures

Figure 7

Voie = Channel

- 5 Constellation produit = Constellation produced  
Domaine de Fourier = Fourier domain  
Domaine temporal = Time domain  
bruit = noise

10 Figure 8

Voie = Channel

Constellation produit = Constellation produced  
Domaine de Fourier = Fourier domain  
Domaine temporal = Time domain

- 15 bruit = noise

Figure 9

Voie = Channel

Constellation produit = Constellation produced

- 20 Domaine de Fourier = Fourier domain  
Domaine temporal = Time domain  
bruit = noise

Figure 10

- 25 Voie = Channel

Constellation produit = Constellation produced  
Domaine de Fourier = Fourier domain  
Domaine temporal = Time domain  
bruit = noise

- 30

Figure 11

Voie = Channel

Constellation produit = Constellation produced  
Domaine de Fourier = Fourier domain

- 35 Domaine temporal = Time domain  
bruit = noise

Figure 12

Voie = Channel  
Constellation produit = Constellation produced  
Domaine de Fourier = Fourier domain  
Domaine temporel = Time domain  
5 bruit = noise

Figure 13

Voie = Channel  
Constellation produit = Constellation produced  
10 Domaine de Fourier = Fourier domain  
Domaine temporel = Time domain  
bruit = noise

Figure 14

15 Voie = Channel  
Constellation produit = Constellation produced  
Domaine de Fourier = Fourier domain  
Domaine temporel = Time domain  
bruit = noise

20

Figure 15

Estimation amplifications = Estimation of  
amplifications  
Estimation dephasages = Estimation of phase-shifts  
25 Estimation Temps de retard = Estimation of delay times  
Estimation residus de porteuse = Estimation of carrier  
residues  
Estimation SNR = Estimation of SNR  
SNR local = Local SNR  
30 SNR global = Global SNR  
Voie 1 locale = Channel 1 local  
Voie 1 globale = Channel 1 global  
Voie 2 locale = Channel 2 local  
Voie 2 globale = Channel 2 global

35