SEQUENTIAL MAP EQUALIZATION OF MIMO CHANNELS AND ITS APPLICATION TO UWB COMMUNICATIONS

Manuel A. Vázquez[*]

University of A Coruña
Departamento de Electrónica e Sistemas
Facultade de Informática, Campus de Elviña s/n
15071 A Coruña, Spain.
E-mail: mvazquez@udc.es

Joaquín Míguez[†]

Universidad Carlos III de Madrid
Departamento de Teoría de la Señal
y Comunicaciones
Avenida de la Universidad 30,
28911 Leganés, Madrid, Spain.
E-mail: jmiguez@ieee.org

ABSTRACT

We introduce new sequential Monte Carlo (SMC) techniques for the maximum *a posteriori* (MAP) equalization of multiple input multiple output (MIMO) wireless channels. SMC methods have been recently proposed to tackle the MIMO equalization problem because of their potential to provide asymptotically optimal performance in terms of bit error rate and their suitability for implementation using parallel hardware. However, the existing algorithms are limited by their high computational complexity relative to the dimensions of the MIMO channel. The SMC equalizers in this paper overcome this drawback by means of a new sampling scheme that constrains the growth of the computational load to be of quadratic order with respect to the channel dimensions. We apply the new algorithms to the equalization of multi-antenna and multiuser (MU) ultra-wide band (UWB) communication systems and provide computer simulation results to illustrate their performance in both scenarios.

1. INTRODUCTION

Multiple input multiple output (MIMO) channels appear in many relevant communication scenarios, such as multiuser [1] and multi-antenna [2] systems. Unfortunately, practical MIMO channel equalization poses several problems. In particular, the computational complexity of maximum *a posteriori* (MAP) data detection grows exponentially with the number of inputs and the length of the channel impulse response (CIR).

Recently, the application of the sequential Monte Carlo (SMC) methodology [3] has been proposed to build quasi-MAP MIMO equalizers with polynomial complexity and amenable to parallel implementation [4, 5, 6]. SMC techniques, also known as particle filtering (PF) algorithms, are based on approximating probability distributions of interest using weighted samples [3], and they are computationally intensive in absolute terms (because usually many samples have to be generated in order to obtain some prescribed level of performance). However, they are also inherently suitable for implementation using parallel hardware and, therefore, hold promise of very high processing speeds. Unfortunately, even the techniques in [5, 6] can be prohibitive for certain classes of MIMO systems because they involve heavy computational tasks (e.g., banks of Kalman filters and successive matrix inversions) with a complexity that grows with L^3, where L is the number of outputs of the MIMO channel.

In this work, we propose two new SMC equalizers for nearly-MAP equalization of MIMO systems that can be implemented with quadratic complexity with respect to the channel dimensions. They are specially suitable for large systems or when there exist stringent real-time requirements (e.g., online equalization). Compared with the methods in [5, 6], the new equalizers substitute the banks of Kalman filters by less complex parallel adaptive channel estimation algorithms that avoid matrix inversions altogether.

The remaining of the paper is organized as follows. In the next section, a general signal model for a frequency-selective MIMO channel is described. This model can be used in a straightforward manner to represent multi-antenna transmission systems and, furthermore, we show that it also comprises a class of multiuser (MU) ultra-wideband (UWB) communication systems [7] as a particular case. In section 3, the standard application of SMC methods to MIMO equalization is discussed. The fundamental ideas behind the proposed SMC equalizers are introduced in Section 4. Then, in Section 5, we apply the new algorithms to the equalization of multi-antenna and MU-UWB communication systems, and provide illustrative computer simulation results. Finally, brief concluding remarks are made in Section 6.

2. SIGNAL MODEL

2.1. Discrete-time model of a general MIMO system

The discrete-time equivalent model of a MIMO transmission system with frequency-selective and time-varying CIR can be written as [5]

$$\mathbf{x}_t = \sum_{i=0}^{m-1} \mathbf{H}_{i,t}\mathbf{b}_{t-i} + \mathbf{u}_t, \quad t \in \mathbb{N}, \tag{1}$$

where $\{\mathbf{H}_{i,t}\}_{i=0}^{m-1}$ is the $L \times N$-dimensional CIR, of length m, $\mathbf{b}_t$ is the $N \times 1$ vector containing the symbols transmitted at time t, $\mathbf{u}_t \sim N(\mathbf{u}_t|\mathbf{0}, \sigma_u^2\mathbf{I}_L)$ is an additive white Gaussian noise (AWGN) process with zero mean and covariance matrix $\sigma_u^2\mathbf{I}_L$ ($\mathbf{I}_L$ is the $L \times L$

[*] M. A. Vázquez acknowledges the support of *Ministerio de Educación y Ciencia* of Spain, (project MIMESIS, ref. TEC2004-06451-C05-01) and *Xunta de Galicia* (project ref. PGIDIT04TIC105008PR).

[†] J. Míguez acknowledges the support of *Ministerio de Educación y Ciencia* of Spain (project DOIRAS, ref. TIC2003-02602), *Comunidad de Madrid* (project PRO-MULTIDIS-CM, ref. S0505/TIC/0223) and the EC (Network of Excellence CRUISE, ref. IST-4-027738).

identity matrix) and $\mathbf{x}_t$ is the $L \times 1$ vector of observations. The symbols are modeled as discrete uniform random variables (r.v.'s) in the alphabet $\mathcal{B}$, hence $\mathbf{b}_t \sim \mathcal{U}(\mathcal{B}^N)$. It is often convenient to use a more compact representation of (1), namely

$$\mathbf{x}_t = \mathbf{H}_t \overline{\mathbf{b}}_t + \mathbf{u}_t, \tag{2}$$

where $\mathbf{H}_t = [\mathbf{H}_{m-1,t} \cdots \mathbf{H}_{0,t}]$ is the $L \times Nm$ overall channel matrix at time t and $\overline{\mathbf{b}}_t = [\mathbf{b}_{t-m+1}^\top \cdots \mathbf{b}_t^\top]^\top$ is an $Nm \times 1$ vector that contains all the symbols involved in the the t-th observation. It is straightforward to verify that either (1) or (2) can be used to model a multi-antenna communication system with N transmitters, L receivers and CIR of length m. Specifically, the sequence of coefficients $\{h_{i,t}^{r,c}\}_{i=0}^{m-1}$, where $h_{i,t}^{r,c}$ is the element at the r-th row and c-th column of $\mathbf{H}_{i,t}$, is the discrete-time equivalent channel, at time t, between the transmit antenna c and the receive antenna r.

The channel variation can be modeled with an autoregressive (AR) process [8], that we assume of first order for simplicity (higher orders are easily handled, except for the notational burden), namely

$$\mathbf{H}_t = \gamma \mathbf{H}_{t-1} + \mathbf{V}_t, \tag{3}$$

where $1 - \epsilon < \gamma < 1$ (for small $\epsilon > 0$) and $\mathbf{V}_t$ is a matrix of independent and identically distributed (i.i.d.) Gaussian random variables with zero mean and variance σ_v^2.

Because of the channel frequency-selectivity, some type of smoothing is needed for reliable data detection. The design of smoothing detectors becomes simpler if we stack together several successive observation vectors, to yield the model

$$\mathbf{x}_{t,a} = \mathbf{H}_{t,a} \mathbf{b}_{t,a} + \mathbf{u}_{t,a}, \tag{4}$$

where $a \geq 1$ is the smoothing lag, $\mathbf{x}_{t,a} = [\mathbf{x}_t^\top \cdots \mathbf{x}_{t+a}^\top]^\top$ is the $L(a+1) \times 1$ vector of stacked observations, $\mathbf{b}_{t,a} = [\mathbf{b}_{t-m+1}^\top \cdots \mathbf{b}_{t+a}^\top]^\top$ has dimensions $N(m+a) \times 1$, $\mathbf{u}_{t,a} = [\mathbf{u}_t^\top \cdots \mathbf{u}_{t+a}^\top]^\top$ and

$$\mathbf{H}_{t,a} = \begin{bmatrix} \mathbf{H}_t(m-1) & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{H}_t(m-2) & \mathbf{H}_{t+1}(m-1) & \cdots & \mathbf{0} \\ \vdots & \mathbf{H}_{t+1}(m-2) & \ddots & \vdots \\ \mathbf{H}_t(0) & \vdots & \ddots & \mathbf{H}_{t+d}(m-1) \\ \vdots & \mathbf{H}_{t+1}(0) & \ddots & \mathbf{H}_{t+d}(m-2) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{H}_{t+d}(0) \\ & & & \end{bmatrix}^\top \tag{5}$$

is the $L(a+1) \times N(m+a)$ stacked channel matrix.

2.2. MIMO model of a MU-UWB transmission system

The discrete-time MIMO model (2) can also be used to represent the signal observed by the receiver in a UWB communication system with time-hopping multiple access. In particular, let us consider the uplink of a system with U users. All of them employ the same pulse waveform, $g(t)$, with support in the interval $[0, T_g)$, for impulse-radio transmission. The distinct code (i.e., the sequence of time hops) assigned to the n-th user is denoted as $\{c_{n,i}\}_{i=0}^{N_f-1}$, where N_f is the code length. The k-th symbol of the n-th user, $b_{n,k}$, is transmitted over N_f frames of duration $T_f >> T_g$, in such a way that the symbol period is $T_b = N_f T_f$ and, during the i-th frame, the user transmits the pulse $g(t - kT_b - iT_f - c_{n,i}T_g)$, where $i \in \{0, ..., N_f - 1\}$, $c_{n,i} \in \{0, ..., N_c - 1\}$ and $N_c = \frac{T_f}{T_g}$.

The signal transmitted by the n-th user,

$$x_n(t) = \sum_{k \in \mathbb{Z}} b_{u,k} \sum_{i=0}^{N_f-1} g(t - kT_b - iT_f - c_{n,i}T_g), \tag{6}$$

$t \in \mathbb{R}$, propagates over a frequency-selective multipath channel with impulse response

$$h_n(t) = \sum_{j=0}^{P-1} \beta_{n,j} \delta(t - jT_g), \tag{7}$$

where $\delta(\cdot)$ is the Dirac delta function. Because of asynchronous transmission, the aggregate received signal has the form

$$\begin{aligned} x(t) \quad = \quad & \sum_{k \in \mathbb{Z}} \sum_{n=1}^{U} b_{n,k} \sum_{i=0}^{N_f-1} \sum_{j=0}^{P-1} \beta_{n,j} \times \\ & \times g(t - kT_b - iT_f - (c_{n,i} + j)T_g - \tau_n) + u(t), \end{aligned} \tag{8}$$

where $\tau_n > 0$ is the symbol delay of the n-th user and $u(t)$ is an AWGN continuous-time process. See [7] for details on the modeling of UWB signals.

The receiver front-end consists of a bank of U frame-rate correlators, hence there are UN_f signal samples available per symbol period, T_b. The output of the n-th correlator for the i-th frame of the k-th symbol can be written as

$$x_{n,i}(k) = \sum_{m \in \mathbb{Z}} \sum_{l=1}^{U} b_{l,m} \sum_{r=0}^{N_f-1} \sum_{j=0}^{P-1} \beta_{l,j} I_{n,i}^{m,l,r,j}(k) + u_{n,i}(k), \tag{9}$$

where $I_{n,i}^{m,l,r,j}(k)$ is the integral

$$\int_0^{T_g} g(\theta) g(\theta + (k-m)T_b + (i-r)T_f + (c_{n,i} - c_{l,r} - j)T_g - \tau_l) d\theta, \tag{10}$$

and the terms of the form $u_{n,i}(k)$ are i.i.d. Gaussian r.v.'s with zero mean and variance σ_u^2.

The amount of inter-symbol interference (ISI) in the system depends on the parameter P (the common length of the CIR for all users) and the maximum symbol delay. It is often realistic to assume the $PT_g + \max_{l \in \{1,...,U\}} \{\tau_l\} < \frac{T_b}{2}$, which implies that the symbols transmitted at time k interfere only with those transmitted at time $k+1$. Under this assumption, we can write

$$\begin{aligned} x_{n,i}(k) \quad = \quad & \sum_{l=1}^{U} \left[b_{l,k-1} q_{n,i+\frac{N_f}{2}}(l, k-1) + b_{l,k} f_{u,i}(l,k) \right] \\ & + u_{n,i}(k), \quad i = 0, ..., \frac{N_f}{2} - 1, \end{aligned} \tag{11}$$

$$x_{n,i} \quad = \quad \sum_{l=1}^{U} b_{l,k} f_{u,i}(l,k) + u_{n,i}(k), \quad i = \frac{N_f}{2}, ..., N_f - 1, \tag{12}$$

where

$$q_{n,i+\frac{N_f}{2}}(l, k-1) \quad = \quad \sum_{r=i+\frac{N_f}{2}}^{N_f-1} \sum_{j=0}^{P-1} \beta_{l,j} I_{n,i}^{k-1,r,l,j}(k), \tag{13}$$

$$f_{n,i}(l,k) \quad = \quad \sum_{r=0}^{i} \sum_{j=0}^{P-1} \beta_{l,j} I_{n,i}^{k,r,l,j}(k). \tag{14}$$

Finally, in order to obtain a representation of the observations of the form (2), let us define the $U \times 1$ vectors $\mathbf{x}_i(k) := [x_{1,i}(k), \ldots, x_{U,i}(k)]^\top$ and $\mathbf{b}_k = [b_{1,k}, \ldots, b_{U,k}]^\top$, the $N_f U \times 1$ vector $\mathbf{x}_k := [\mathbf{x}_0^\top(k), \ldots, \mathbf{x}_{N_f-1}^\top(k)]^\top$, and the $U \times U$ composite-channel matrices $\mathbf{Q}_{i+\frac{N_f}{2}}(k-1)$ and $\mathbf{F}_i(k)$, given elemen-wise as $[\mathbf{Q}_{i+\frac{N_f}{2}}(k-1)]_{r,c} := q_{r,i+\frac{N_f}{2}}(c, k-1)$, $[\mathbf{F}_i(k)]_{r,c} := f_{r,i}(c,k)$, where $r, c \in \{1, ..., U\}$ are row and column indices. We can write

$$\mathbf{x}_k = \mathbf{H}_k \overline{\mathbf{b}}_k + \mathbf{u}_k, \tag{15}$$

where

$$\mathbf{H}_k := \begin{bmatrix} \mathbf{Q}_{\frac{N_f}{2}}(k-1) & \mathbf{F}_0(k) \\ \vdots & \vdots \\ \mathbf{Q}_{N_f-1}(k-1) & \mathbf{F}_{\frac{N_f}{2}-1}(k) \\ \mathbf{0} & \mathbf{F}_{\frac{N_f}{2}}(k) \\ \vdots & \vdots \\ \mathbf{0} & \mathbf{F}_{N_f-1}(k) \end{bmatrix}, \quad \overline{\mathbf{b}}_k := \begin{bmatrix} \mathbf{b}_{k-1} \\ \mathbf{b}_k \end{bmatrix} \tag{16}$$

and $\mathbf{u}_k = [u_{1,0}(k), \ldots, u_{1,N_f-1}(k), \ldots, u_{U,N_f-1}(k)]^\top$.

3. STANDARD SMC EQUALIZATION OF MIMO CHANNELS

In this section we summarize the standard approach to MIMO channel equalization using SMC algorithms, with special attention to its computational complexity. Similar material can be found in [9, 10, 3, 5]. We assume the general model (2) for our derivations (application to the MU-UWB model (15) is straightforward).

3.1. Sequential Importance Sampling

Most particle filtering methods rely upon the principle of importance sampling (IS) [3] for building an empirical approximation of a desired pdf[1], say $p(x)$, by drawing samples from a different distribution, known as *importance function* or *proposal pdf* and denoted $\pi(x)$. These samples are then assigned appropriate normalized *importance* weights, i.e.,

$$x^{(i)} \sim \pi(x) \quad \text{and} \quad w^{(i)} \propto \frac{p(x^{(i)})}{\pi(x^{(i)})},$$

where $\sum_{i=1}^M w^{(i)} = 1$, M being the number of particles. In order to detect the transmitted symbols, it is natural to aim at the approximation of the *a posteriori* marginal pdf of the data, $p(\mathbf{b}_{0:t}|\mathbf{x}_{0:t})$, which contains all relevant statistical information for the optimal (Bayesian) estimation of $\mathbf{b}_{0:t}$. In turn, an importance function of the form $\pi(\mathbf{b}_{0:t}|\mathbf{x}_{0:t})$ is needed.

One of the most appealing features of the particle filtering approach is its potential for online processing. Indeed, the IS principle can be sequentially applied by exploiting the recursive decomposition of the posterior distribution

$$p(\mathbf{b}_{0:t}|\mathbf{x}_{0:t}) \propto p(\mathbf{x}_t|\mathbf{b}_{0:t}, \mathbf{x}_{0:t-1}) p(\mathbf{b}_{0:t-1}|\mathbf{x}_{0:t-1}), \tag{17}$$

which is easily derived by taking into account the *a priori* uniform distribution of the symbols, and an adequate importance function that can be factored as

$$\pi(\mathbf{b}_{0:t}|\mathbf{x}_{0:t}) = \pi(\mathbf{b}_t|\mathbf{b}_{0:t-1}, \mathbf{x}_{0:t}) \pi(\mathbf{b}_{0:t-1}|\mathbf{x}_{0:t-1}). \tag{18}$$

The recursive algorithm that combines the IS principle and decompositions (17) and (18) to build a discrete random measure that approximates the posterior pdf is called sequential importance sampling (SIS) [3]. Let $\Omega_t = \left\{\mathbf{b}_{0:t}^{(i)}, w_t^{(i)}\right\}_{i=1}^M$ denote the discrete measure at time t, where M is the number of particles. The desired pdf is approximated as

$$\hat{p}(\mathbf{b}_{0:t}|\mathbf{x}_{0:t}) = \sum_{i=1}^M \delta_i(\mathbf{b}_{0:t}) w_t^{(i)}, \tag{19}$$

where $\delta_i(\mathbf{b}_t) = \delta(\mathbf{b}_t - \mathbf{b}_t^{(i)})$ is the Dirac delta function. When a new observation is collected at time $t+1$, the SIS algorithm proceeds through the following steps to recursively compute Ω_{t+1},

1. Importance sampling: $\mathbf{b}_{t+1}^{(i)} \sim \pi(\mathbf{b}_{t+1}|\mathbf{b}_{0:t}^{(i)}, \mathbf{x}_{0:t+1})$.
2. Weight update: $\tilde{w}_{t+1}^{(i)} = w_t^{(i)} \frac{p(\mathbf{x}_{t+1}|\mathbf{b}_{0:t+1}^{(i)}, \mathbf{x}_{0:t})}{\pi(\mathbf{b}_{t+1}^{(i)}|\mathbf{b}_{0:t}^{(i)}, \mathbf{x}_{0:t+1})}$
3. Weight normalization: $w_t^{(i)} = \frac{\tilde{w}_t^{(i)}}{\sum_{k=1}^N \tilde{w}_t^{(k)}}$

It is straightforward to obtain data estimates from the approximate pdf $\hat{p}(\mathbf{b}_{0:t}|\mathbf{x}_{0:t})$. In particular, the marginal MAP symbol detector is

$$\hat{\mathbf{b}}_t^{map} = \arg\max_{\mathbf{b}_t} \left\{ \sum_{i=1}^M \delta(\mathbf{b}_t - \mathbf{b}_t^{(i)}) w_t^{(i)} \right\}, \tag{20}$$

which amounts to selecting the particle with the highest accumulated weight (note that some particles can be replicated).

One major problem in the practical implementation of the SIS algorithm is that after few time steps most of the particles have importance weights with negligible values (very close to zero). The common solution to this problem is to *resample* the particles. Resampling is an algorithmic step that stochastically discards particles with small weights while replicating those with significant weight. In this paper, we consider only the conceptually simplest resampling scheme, that generates a set of M new and equally weighted particles, $\{\mathbf{b}_{0:t}^{(i)}, 1/M\}_{i=1}^M$, by drawing from the discrete probability distribution $p_{rsp}(\mathbf{b}_{0:t}^{(i)}) = w_t^{(i)}$. Unfortunately, resampling may cause major difficulties for the implementation of SMC algorithms with parallel hardware, because it requires the joint processing of all the particle weights. A discussion of the implementation problems associated to some common resampling schemes, as well as techniques to tackle them, can be found in [11].

3.2. Delayed Sampling

The performance of the SIS algorithm considerably depends on the choice of importance function, π. However, even the optimal choice of π can yield a poor performance when the channel is frequency selective, even for small values of the CIR length, m, as shown in [5]. Detection in dispersive channels usually requires some smoothing (i.e., to exploit posterior observations, $\mathbf{x}_{0:t+a}$, where $0 < a \leq m-1$ is a smoothing lag) in order to detect $\mathbf{b}_t$. In the context of particle filtering, smoothing is also referred to as *delayed sampling* [9] because particle $\mathbf{b}_t^{(i)}$ cannot be drawn until $\mathbf{x}_{t+a}$ is observed.

[1] We note that any probability mass function can be expressed as a density using sums of Dirac delta functions.

The optimal smoothing importance pdf is

$$\begin{aligned}&\pi(\mathbf{b}_t|\mathbf{b}_{0:t-1},\mathbf{x}_{0:t+a}) = p(\mathbf{b}_t|\mathbf{b}_{0:t-1},\mathbf{x}_{0:t+a})\\ &\propto \textstyle\sum_{\tilde{\mathbf{b}}_{t+1:t+a}\in\mathcal{B}^{Na}} \prod_{k=0}^{a} p\left(\mathbf{x}_{t+k}|\mathbf{b}_{0:t},\tilde{\mathbf{b}}_{t+1:t+k},\mathbf{x}_{0:t+k-1}\right),\end{aligned} \tag{21}$$

In order to sample the importance function of (21), it is necessary to evaluate the likelihoods $p(\mathbf{x}_{t+k}|\mathbf{b}_{0:t},\mathbf{x}_{0:t-1})$, for $k = 0, ..., a$ and for each possible value of the sequence of vectors $\mathbf{b}_{t:t+a}$, normalize them and then draw from the resulting discrete distribution. Unfortunately, there are $|\mathcal{B}|^{N(a+1)}$ possibilities for $\mathbf{b}_{t:t+a}$ and the evaluation of each likelihood involves running one step of a Kalman filter which, in turn, has a computational complexity $\mathcal{O}(L^3)$ because of required matrix inversions. Therefore, the complexity of the algorithm grows exponentially with the number of transmit antennas and the smoothing lag, i.e., it is $\mathcal{O}(|\mathcal{B}|^{N(a+1)})$.

4. NEW SMC-MAP EQUALIZERS

The SMC equalizer based on the optimal delayed importance function is limited by its practically intractable computational complexity. In [5, 6], new SMC smoothing schemes were proposed that avoid the exponential growth of complexity, but they still require to run banks of Kalman filters, as well as additional matrix inversions, that involve $\mathcal{O}\left((L(a+1))^3\right)$ operations. For many MIMO systems, this is prohibitive, e.g., assuming the simplest case $a = 1$, the complexity of these equalizers for the MU-UWB system of Section 2.2 is $\mathcal{O}((UN_f)^3)$, where the code length N_f can be very large.

In order to drastically reduce this stringent computational requirements, we propose a scheme to directly approximate, using samples, the joint smoothing distribution of the sequence of symbol vectors and channel matrices given the observations, i.e., $p(\mathbf{b}_{0:t},\mathbf{H}_{0:t}|\mathbf{x}_{1:t+a})$, instead of the marginal *a posteriori* pdf of (17). Assume the general model of (2) for the observations. The joint posterior *filtering* pdf admits the straightforward decomposition

$$\begin{aligned}p(\mathbf{b}_{0:t+a},\mathbf{H}_{0:t+a}|\mathbf{x}_{1:t+a}) \quad &\propto \quad p(\mathbf{x}_{t:t+a}|\mathbf{b}_{t-m+1:t+a},\mathbf{H}_{t:t+a})\\ &\times \quad p(\mathbf{H}_{t:t+a}|\mathbf{H}_{t-1})\\ &\times \quad p(\mathbf{b}_{0:t-1},\mathbf{H}_{0:t-1}|\mathbf{x}_{1:t-1}).\end{aligned} \tag{22}$$

Notice that $p(\mathbf{x}_{t:t+a}|\mathbf{b}_{t-m+1:t+a},\mathbf{H}_{t:t+a})$ and $p(\mathbf{H}_{t:t+a}|\mathbf{H}_{t-1})$ are Gaussian pdf's, straightforward to evaluate without the need of Kalman filtering. Assume also a suitable proposal pdf of the form

$$\begin{aligned}&\pi_t(\mathbf{b}_{t:t+a},\mathbf{H}_{t:t+a}|\mathbf{b}_{0:t-1}^{(i)},\mathbf{H}_{0:t-1}^{(i)},\mathbf{x}_{1:t+a}) \propto\\ &\pi_t(\mathbf{b}_{t:t+a}|\mathbf{H}_{t:t+a})\pi_t(\mathbf{H}_{t:t+a}),\end{aligned} \tag{23}$$

with $\pi_t(\mathbf{b}_{t:t+a}|\mathbf{H}_{t:t+a})$ and $\pi_t(\mathbf{H}_{t:t+a})$ to be specified later, and the weight update rule

$$\begin{aligned}w_{t+a}^{(i)} \quad &\propto \quad w_{t+a-1}\frac{p(\mathbf{x}_{t:t+a}|\mathbf{b}_{t-m+1:t+a}^{(i)},\mathbf{H}_{t:t+a}^{(i)})}{\pi_t(\mathbf{b}_{t:t+a}^{(i)}|\mathbf{H}_{t:t+a}^{(i)})}\\ &\times\frac{p(\mathbf{H}_{t:t+a}^{(i)}|\mathbf{H}_{t-1}^{(i)})}{\pi_t(\mathbf{H}_{t:t+a}^{(i)})}.\end{aligned} \tag{24}$$

The use of (23) and (24) yields a new set of weighted particles, $\tilde{\Omega}_{t+a} = \left\{\left(\mathbf{b}_{0:t+a}^{(i)},\mathbf{H}_{0:t+a}^{(i)}\right), w_{t+a}^{(i)}\right\}_{i=1}^{M}$, and the approximation

$$p(\mathbf{b}_{0:t+a},\mathbf{H}_{0:t+a}|\mathbf{x}_{1:t+a}) \approx \sum_{i=1}^{M} w_{t+a}^{(i)}\delta_i(\mathbf{b}_{0:t+a})\delta_i(\mathbf{H}_{0:t+a}). \tag{25}$$

Integrating (25) over $\mathbf{b}_{t+1:t+a}$ and $\mathbf{H}_{t+1:t+a}$, yields an estimate of the desired joint smooting pdf,

$$\begin{aligned}p(\mathbf{b}_{0:t},\mathbf{H}_{0:t}|\mathbf{x}_{1:t+a}) \quad &\approx \quad \int\int\sum_{i=1}^{M} w_{t+a}^{(i)}\delta_i(\mathbf{b}_{0:t+a})\delta_i(\mathbf{H}_{0:t+a})\\ &= \quad \sum_{i=1}^{M} w_{t+a}^{(i)}\delta_i(\mathbf{b}_{0:t})\delta_i(\mathbf{H}_{0:t}).\end{aligned} \tag{26}$$

Therefore, it is only necessary to keep the weighted particles up to time t, $\Omega_{t+a} = \left\{\left(\mathbf{b}_{0:t}^{(i)},\mathbf{H}_{0:t}^{(i)}\right), w_{t+a}^{(i)}\right\}_{i=1}^{M}$, and apply (23), (24) and (26) sequentially, with resampling steps when needed. Approximate MAP, smooth symbol estimates are computed as

$$\hat{\mathbf{b}}_t^{map} = \arg\max_{\mathbf{b}_t}\left\{\sum_{i=1}^{M}\delta(\mathbf{b}_t - \mathbf{b}_t^{(i)})w_{t+a}^{(i)}\right\}. \tag{27}$$

4.1. Channel Sampling Scheme

Although the weight update equation (24) enables us to circumvent the use of the Kalman filter (KF) banks in [9, 5, 6], we still need to design a proposal pdf π_t that avoids matrix inversions and any other "heavy" computations. With that aim, we propose to use a bank of (simple) adaptive channel estimators that play the same role as the KF in [5], but with less stringent requirements. A similar idea was applied, for single-user spread spectrum systems, in [12]. In this paper, we will consider both least mean squares (LMS) and recursive least squares (RLS) [13] channel estimation algorithms, to be briefly described in Sections 4.1.1 and 4.1.2, respectively.

Independently of the channel estimation method, at time t there is an available estimate, $\hat{\mathbf{H}}_{t-1}^{(i)}$, for each $i \in \{1, ..., M\}$. Taking into account the AR model of the channel variation, we propose to draw $\mathbf{H}_t^{(i)}$ from a Gaussian proposal pdf with mean $\gamma\hat{\mathbf{H}}_{t-1}^{(i)}$ and diagonal covariance matrix $\sigma_H^2\mathbf{I}$, where σ_H^2 is a design parameter. The remaining channel samples, $\mathbf{H}_{t+1:t+a}^{(i)}$, are then drawn using the AR model. In summary,

$$\begin{aligned}\mathbf{H}_{t:t+a}^{(i)} \quad &\sim \quad \pi_t(\mathbf{H}_{t:t+a}) = N(\mathbf{H}_t|\gamma\hat{\mathbf{H}}_{t-1}^{(i)},\sigma_H^2\mathbf{I})\\ &\times\prod_{k=1}^{a} N(\mathbf{H}_{t+k}|\gamma\mathbf{H}_{t+k-1},\sigma_v^2\mathbf{I}).\end{aligned} \tag{28}$$

Given $\mathbf{H}_{t:t+a}^{(i)}$, we can draw the new symbol vector $\mathbf{b}_t^{(i)}$ (see Section 4.2 for details) and then update the bank of adaptive channel estimators, to yield $\hat{\mathbf{H}}_t^{(i)}$, $i = 1, ..., M$.

4.1.1. *LMS Channel Estimation*

Consider the minimum mean square error (MMSE) estimation of the channel given $\mathbf{b}_{0:t}^{(i)}$, $i \in \{1, ..., M\}$, and $\mathbf{x}_{1:t}$, i.e.,

$$\hat{\mathbf{H}}_t^{(i)} = \arg\min_{\mathbf{H}_t}\left\{E\left[\left\|\mathbf{x}_t - \mathbf{H}_t\overline{\mathbf{b}}_t^{(i)}\right\|^2\right]\right\}. \tag{29}$$

The simplest way to adaptively solve (29) is the LMS algorithm [13], which takes the form

$$\hat{\mathbf{H}}_t^{(i)} = \hat{\mathbf{H}}_{t-1}^{(i)} - \mu\left(\hat{\mathbf{H}}_{t-1}^{(i)}\overline{\mathbf{b}}_t^{(i)} - \mathbf{x}_t\right)\overline{\mathbf{b}}_t^{(i)^H}, \tag{30}$$

where $\mu << 1$ is a step-size parameter.

4.1.2. RLS Channel Estimation

The LMS algorithm (30) has linear computational complexity, but it usually exhibits a slow convergence rate and poor tracking capabilities. To avoid these well-known drawbacks it is convenient to use the exponentially-weighted RLS algorithm [13]. The channel estimator is

$$\hat{\mathbf{H}}_t^{(i)} = \arg\min_{\mathbf{H}} \left\{ \sum_{k=0}^{t} \lambda^{t-k} \left\| \mathbf{x}_k - \mathbf{H}\overline{\mathbf{b}}_k \right\|^2 \right\}, \quad t = 1, 2, \ldots, \tag{31}$$

where $0 < \lambda < 1$ is a forgetting factor. The sequence of problems (31) can be recursively solved using the RLS algorithm, which consists of the following two steps.

1. Initialization,

$$\begin{aligned} \mathbf{R}_0^{(i)^{-1}} &\propto \mathbf{I}_{Nm}, & (32)\\ \hat{\mathbf{H}}_0^{(i)} &= \mathbf{0}. & (33) \end{aligned}$$

2. Recursive update,

$$\mathbf{g}_t^{(i)^H} = \frac{\lambda^{-1}\overline{\mathbf{b}}_t^{(i)^H}\mathbf{R}_{t-1}^{(i)^{-1}}}{1+\lambda^{-1}\overline{\mathbf{b}}_t^{(i)^H}\mathbf{R}_{t-1}^{(i)^{-1}}\overline{\mathbf{b}}_t^{(i)}} \tag{34}$$

$$\hat{\mathbf{H}}_t^{(i)} = \hat{\mathbf{H}}_{t-1}^{(i)} + \left(\mathbf{x}_t - \hat{\mathbf{H}}_{t-1}^{(i)}\overline{\mathbf{b}}_t^{(i)}\right)\mathbf{g}_t^{(i)^H} \tag{35}$$

$$\mathbf{R}_t^{(i)^{-1}} = \lambda^{-1}\mathbf{R}_{t-1}^{(i)^{-1}}\left(\mathbf{I}_{Nm} + \overline{\mathbf{b}}_t^{(i)}\mathbf{g}_t^{(i)^H}\right). \tag{36}$$

The complexity of the resulting equalizer is linear with respect to LNm. Since, normally, $L \geq N$, it becomes $\mathcal{O}(N^2)$.

4.2. Data Sampling Scheme

When the channel sample, $\mathbf{H}_{t:t+d}^{(i)}$, $i \in \{1, ..., M\}$, is available, we build a data proposal pdf based on linear MMSE detection, as suggested in [6], but avoiding the computation of inverse matrices. In particular, we exploit the matrix inversion lemma [13] to recursively approximate the inverse of the autocorrelation matrix

$$\mathbf{R}_{t,x}^{-1} = \left(\frac{1}{t}\sum_{n=0}^{t}\mathbf{x}_{t,a}\mathbf{x}_{t,a}^H\right)^{-1} \tag{37}$$

as

$$\hat{\mathbf{R}}_{0,x}^{-1} \propto \mathbf{I}_{L(a+1)}, \quad \hat{\mathbf{R}}_{t,x}^{-1} = \alpha^{-1}\left(\mathbf{I}_{L(a+1)} - \mathbf{g}_{t,a}\mathbf{x}_{t,a}^H\right)\hat{\mathbf{R}}_{t-1,x}^{-1}, \tag{38}$$

where $0 < \alpha < 1$ is a forgetting factor and $\mathbf{g}_{t,a} = \frac{\alpha^{-1}\hat{\mathbf{R}}_{t-1}^{-1}\mathbf{x}_{t,a}}{1+\alpha^{-1}\mathbf{x}_{t,a}^H\hat{\mathbf{R}}_{t-1}^{-1}\mathbf{x}_{t,a}}$ is a gain vector. A bank of $N(a+1)$ MMSE linear filters is built,

$$\mathbf{W}_{t,a}^{(i)} = \sigma_b^2 t\hat{\mathbf{R}}_{t,x}^{-1}\mathbf{H}_{t,a}^{(i)}\mathbf{E}, \tag{39}$$

for $i = 1, ..., M$, where $\mathbf{W}_{t,a}^{(i)}$ has dimensions $L(a+1) \times N(a+1)$, σ_b^2 is the symbol power and $\mathbf{E} = \begin{bmatrix} \mathbf{0}_{N(m-1)\times N(a+1)} \\ \mathbf{I}_{N(a+1)} \end{bmatrix}$, and $N(a+1)$ soft data estimates are computed,

$$\mathbf{y}_{t,a}^{(i)} = \mathbf{W}_{t,a}^{(i)^H}\mathbf{x}_{t,a}. \tag{40}$$

Let $y_{j,t,a}^{(i)}$ denote the j-th element in $\mathbf{y}_{t,a}^{(i)}$, let $b_{l,t}$ be the l-th symbol in $\mathbf{b}_t$ and let $j = Nk + q$ for integers $k, q \geq 0$. Then, $y_{j,t,a}^{(i)}$ is an estimate of $b_{q,t+k}$. If the symbols are binary, $b_{q,t} \in \{\pm 1\}$ (extension to higher order constellations is straightforward), we can assign probabilities $\pi_{+1,q,t}^{(i)} \propto \exp\{-\frac{1}{\sigma_y^2}|y_{j,q,t} - 1|^2$ (where σ_y^2 is a design parameter) and $\pi_{-1,q,t}^{(i)} = 1 - \pi_{+1,q,t}^{(i)}$, and draw a sample $b_{q,t}^{(i)}$ accordingly. Repeating this process for all symbols from time t to time $t + a$ we obtain the desired sample $\mathbf{b}_{t:t+a}^{(i)}$. The evaluation of $\pi_t(\mathbf{b}_{t:t+a}^{(i)}|\mathbf{H}_{t:t+a}^{(i)})$ is carried out by adequately multiplying the probabilities $\pi_{\pm 1,q,t}^{(i)}$ for $q = 1, ..., N$ and $t, ..., t + a$.

4.3. Summary

The proposed algorithms are outlined in Table 1. A training sequence, consisting of known symbols $\mathbf{b}_{0:K-1}$, is assumed for a better initialization of the algorithms.

```
Initialization.
  Let $\mathbf{b}_{0:K-1}$ be known (training) data.
  Apply eq. (38) to compute $\hat{\mathbf{R}}_{x,K-1}^{-1}$.
  RLS channel estimation:
    Apply eqs. (32)-(36) to compute $\hat{\mathbf{H}}_{K-1}$ and $\mathbf{R}_{K-1}^{-1}$
    (neglect superscript $^{(i)}$) using $\mathbf{b}_{0:K-1}$.
  LMS channel estimation:
    Apply (30) to compute $\hat{\mathbf{H}}_{K-1}$ (neglect superscript $^{(i)}$)
    using $\mathbf{b}_{0:K-1}$.
  Set $\mathbf{b}_{K-1}^{(i)} = \mathbf{b}_{K-1}$ and $\hat{\mathbf{H}}_{K-1}^{(i)} = \hat{\mathbf{H}}_{K-1}$, $\mathbf{H}_{K-1}^{(i)} = \hat{\mathbf{H}}_{K-1}$
    for $i = 1, ..., M$.
  Set $\mathbf{R}_{K-1}^{(i)^{-1}} = \mathbf{R}_{K-1}^{-1}$ for $i = 1, ..., M$ if RLS channel
    estimation is selected.
  Set $w_{K-1}^{(i)} = 1/M$ for $i = 1, ..., M$.

Recursive step: $t \geq K$
  1. Channel sampling: draw $\mathbf{H}_{t:t+a}^{(i)} \sim \pi_t(\mathbf{H}_{t:t+a})$ using
    eq. (28), $i = 1, ..., M$.
  2. Data sampling: update $\hat{\mathbf{R}}_{x,t}^{-1}$ and:
    Compute $\mathbf{y}_{t,a}^{(i)} = \mathbf{W}_{t,a}^{(i)^H}\mathbf{x}_{t,a}$.
    Draw $\mathbf{b}_{t:t+a}^{(i)}$ as described in Section 4.2.
  3. Update the weights using eq. (24).
  4. Channel update:
    LMS: use eq. (30) to compute $\hat{\mathbf{H}}_t^{(i)}$.
    RLS: use eqs. (34)-(36) to compute $\hat{\mathbf{H}}_t^{(i)}$, $\mathbf{R}_t^{(i)^{-1}}$.
  5. Estimation:
    $\hat{\mathbf{b}}_t = \arg\max_{\mathbf{b}_t}\left\{\sum_{i=1}^{M}\delta(\mathbf{b}_t - \mathbf{b}_t^{(i)})w_{t+a}^{(i)}\right\}$
    $\hat{\mathbf{H}}_t = \sum_{i=1}^{M}\hat{\mathbf{H}}_t^{(i)}w_{t+a}^{(i)}$
  6. Resampling.
```

Table 1. Proposed MIMO SMC equalization scheme.

5. SIMULATIONS

5.1. A 3×2 Multi-Antenna System

Consider a simple system with $N = 2$ transmitting antennas, $L = 3$ receiving antennas and CIR length $m = 2$. The parameters of the channel AR process are $\gamma = 1 - 10^{-5}$ and $\sigma_v^2 = 10^{-4}$. Also assume a BPSK modulation format, thus $\mathcal{B} = \{\pm 1\}$, and burst data transmission in blocks of 300 symbol vectors (i.e., 600 binary symbols overall), including a training sequence of $K = 30$ symbol vectors.

Within this simulation setup, we have compared the optimal smoothing SMC equalizer described in Section 3.2 (labeled "D-SIS opt") with the two low-complexity SMC smoothers proposed in this paper (labeled "LMS-D-SIS" and "RLS-D-SIS", depending on the type of adaptive channel estimator used). The smoothing lag is $a = 1$ for the three equalizers. The plots described below have been obtained from the simulation of 40 independent simulations (one data block per simulation).

Figure 1 (upper) depicts the Bit Error Rate (BER) of the different equalizers for several SNR values, when the number of particles generated by the SMC algorithms is $M = 30$. The curve labeled "MLSD" shows the performance of the maximum likelihood sequence detector (MLSD) implemented via the Viterbi algorithm, with perfect knowledge of the time-varying CIR, and serves as a reference for performance. It can be seen that the "RLS-D-SIS" equalizer attains practically optimal performance up to SNR=9 dB. The "LMS-D-SIS" equalizer has an approximately constant loss of ≈ 1 dB for the whole range of SNR values, but has the advantage of its greater simplicity.

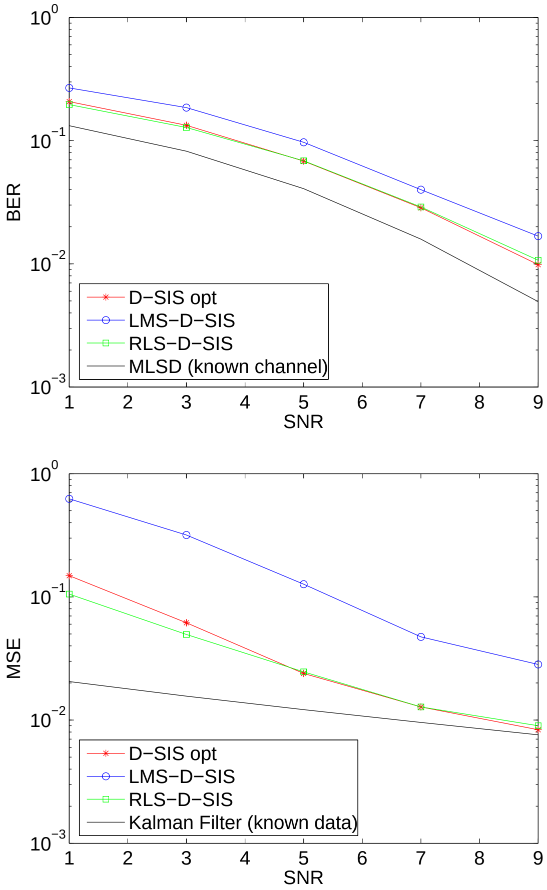

Fig. 1. **Upper:** BER of the SMC equalizers and the MLSD for several values of the SNR (M=100 particles). **Lower:** Approximate MSE, in channel estimation, of the SMC equalizers and the Kalman filter for several values of the SNR (M=100 particles).

Similar results are obtained in terms of the normalized mean square error (MSE) of the channel estimates, as shown in Figure 1 (lower). The normalized MSE is defined as

$$MSE(t) = \frac{1}{S}\sum_{s=1}^{S} \frac{\text{Trace}\left[\left(\hat{\mathbf{H}}_t(s) - \mathbf{H}_t(s)\right)\left(\hat{\mathbf{H}}_t(s) - \mathbf{H}_t(s)\right)^\top\right]}{\text{Trace}\left(\mathbf{H}_t(s)\mathbf{H}_t^\top(s)\right)}, \tag{41}$$

where $S = 40$ is the number of simulation trials and $\hat{\mathbf{H}}_t(s)$ is the estimate of $\mathbf{H}_t(s)$ provided by an equalizer in the s-th simulation. We observe that both the "D-SIS opt" and "RLS-D-SIS" equalizers attain nearly equal MSE for the range of considered SNR values, while the "LMS-D-SIS" method suffers from a performance loss.

5.2. A Simple MU-UWB system

For the next set of computer experiments, consider a MU-UWB communication system with $U = 2$ users, $N_f = 16$ frames per symbol, $N_c = 16$ pulse slots per frame (i.e., $T_f = 16T_g$, with $T_g = 1$ for convenience) and $P = 4N_c = 64$ pulse-rate channel coefficients per user ($\beta_{n,j}$ with $n = 1, 2$ and $j = 0, ..., 63$, according to the notation in Section 2.2). Each user transmits data in bursts of 1000 BPSK-modulated symbols, including a leading traning sequence of length $K = 100$.

For the simulations, we have assumed that the non-zero coefficients in the overall channel matrix $\mathbf{H}_k$, $k = 0, ..., 999$, of (15) are time-varying. The variation model is of the AR type, with the fundamental parameters $\beta_{n,j}$ being randomly and independently generated for each simulation trial ($\beta_{n,j} \sim N(\beta|0, 0.01)$), and the coefficients of $\mathbf{H}_k$ being updated as

$$\mathbf{H}_k(1) = \gamma\mathbf{H}_{k-1}(1) + \mathbf{U}_k(1), \tag{42}$$

$$\mathbf{H}_k(0) = \gamma\mathbf{H}_{k-1}(0) + \mathbf{U}_k(0), \tag{43}$$

where $\gamma = 1 - 10^{-5}$,

$$\mathbf{H}_k(1) = \begin{bmatrix} \mathbf{Q}_{\frac{N_f}{2}}(k-1) & \mathbf{F}_0(k) \\ \vdots & \vdots \\ \mathbf{Q}_{N_f-1}(k-1) & \mathbf{F}_{\frac{N_f}{2}-1}(k) \end{bmatrix}, \tag{44}$$

$$\mathbf{H}_k(0) = \begin{bmatrix} \mathbf{F}_{\frac{N_f}{2}}(k) \\ \vdots \\ \mathbf{F}_{N_f-1}(k) \end{bmatrix} \tag{45}$$

and $\mathbf{U}_k(1)$ and $\mathbf{U}_k(0)$ are $U\frac{N_f}{2} \times 2U$ and $U\frac{N_f}{2} \times U$ matrices, respectively, with random i.i.d. elements distributed according to a $N(u|0, 10^{-6})$ Gaussian pdf.

We have applied the "RLS-D-SIS" algorithm to the equalization of the resulting MIMO channel. One peculiarity of the MU-UWB model is that, because of the structure of $\mathbf{H}_k$ (namely, because of the $U\frac{N_f}{2} \times U$ lower-left zero submatrix) it is convenient to run independent RLS algorithms to estimate $\mathbf{H}_k(0)$ and $\mathbf{H}_k(1)$, hence the "RLS-D-SIS" equalizer with M particles requires a bank of $2M$ RLS procedures.

Figure 2 shows the normalized MSE (defined according to (41)) of the channel estimates attained by the proposed SMC equalizer as a block of data is processed. As a reference, the MSE of the channel estimates computed via a genie-aided RLS algorithm with knowledge of the transmitted symbols, $\mathbf{b}_{0:999}$, is also shown. It can be seen that the proposed equalizer attains practically optimal

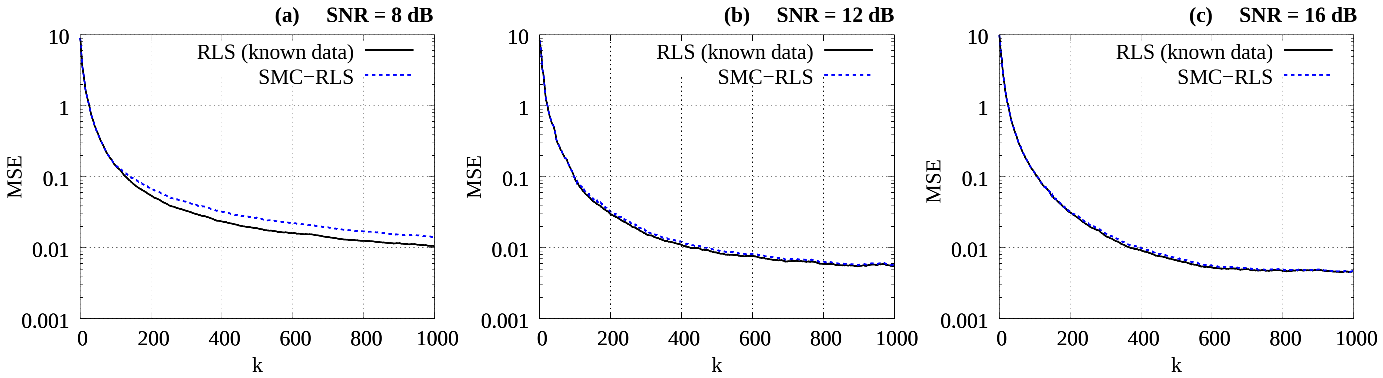

Fig. 2. Evolution of the normalized MSE for **(a)** SNR=8 dB, **(b)** SNR=12 dB and **(c)** SNR=16 dB. The SMC equalizer employs $M = 50$ particles.

channel estimation for SNR$\geq$ 12 dB (and only a small loss for SNR=8 dB). The number of particles generated by the particle filtering algorithm was set to $M = 50$ and the presented plots are the result of averaging the errors in $S = 15$ independent simulation trials.

The average BER attained by the SMC equalizer in these trials, compared to the BER of a MLSD with perfect knowledge of the $\mathbf{H}_k$ matrices, is shown in Table 2.

	SNR=8 dB	12 dB	16 dB
MLSD	0.038	0.015	0.001
RLS-D-SIS	0.059	0.031	0.003

Table 2. BER of the “RLS-D-SIS” equalizer ($M = 50$ particles) and the genie-aided MLSD detector with perfect knowledge of the channel matrices (rounded to within ± 0.001).

6. CONCLUSIONS

Existing particle filtering methods for MIMO channel equalization suffer from severe limitations because of their high computational requirements. In this paper we have introduced two low complexity sampling schemes that perform $\mathcal{O}(N^2)$ operations per particle, N being the number of inputs to the MIMO channel. The proposed equalizers are particularly suitable for implementation using parallel hardware and application in scenarios where either online detection is required or large dimensional MIMO channels need to be estimated. We have illustrated the performance of the proposed methods by computer simulations of a 2-input, 3-output multi-antenna system and a multiuser UWB transmission system.

7. REFERENCES

[1] S. Verdú, *Multiuser Detection*, Cambdridge University Press, Cambridge (UK), 1998.

[2] G. G. Raleigh and J. M. Cioffi, “Spatio-temporal coding for wireless communication,” *IEEE Transactions Communications*, vol. 46, no. 3, pp. 357–366, March 1998.

[3] P. M. Djurić, J. H. Kotecha, J. Zhang, Y. Huang, T. Ghirmai, M. F. Bugallo, and J. Míguez, “Particle filtering,” *IEEE Signal Processing Magazine*, vol. 20, no. 5, pp. 19–38, September 2003.

[4] J. Zhang, Y. Huang, and P. M. Djurić, “Multiuser detection with particle filtering,” in *Proceedings of XI EUSIPCO*, September 2002.

[5] M. A. Vázquez and J. Míguez, “A complexity-constrained particle filtering algorithm for MAP equalization of frequency-selective MIMO channels,” in *Proceedings of the IEEE 30th ICASSP*, March 2005.

[6] M. A. Vázquez, M. F. Bugallo, and J. Míguez, “Novel SMC techniques for blind equalization of flat-fading MIMO channels,” in *Proceedings of the IEEE 61st VTC*, May 2005.

[7] L. Yang and G. B. Giannakis, “Ultra-wideband communications: an idea whose time has come,” *IEEE Signal Processing Magazine*, vol. 21, no. 6, pp. 26–54, March-June 2004.

[8] C. Komnikakis, C. Fragouli, A. H. Sayeed, and R. D. Wesel, “Multi-input multi-output fading channel tracking and equalization using kalman estimation,” *IEEE Transactions Signal Processing*, vol. 50, no. 5, pp. 1065–1076, May 2002.

[9] J. Zhang and P. M. Djurić, “Joint estimation and decoding of space-time trellis codes,” *Journal of Applied Signal Processing*, vol. 3, pp. 305–315, 2002.

[10] D. Guo and X. Wang, “Blind detection in MIMO systems via sequential Monte Carlo,” *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 3, pp. 464–473, April 2003.

[11] M. Bolić, P. M. Djurić, and S. Hong, “Resampling algorithms for particle filters: A computational complexity perspective,” *EURASIP Journal on Applied Signal Processing*, vol. 2004, no. 15, pp. 2267–2277, November 2004.

[12] T. Bertozzi, D. Le Ruyet, C. Panazio, and H. Vu-Thien, “Channel tracking using particle filtering in unresolvable multipath environments,” *EURASIP Journal on Applied Signal Processing*, vol. 2004, no. 15, pp. 2328–2338, November 2004.

[13] S. Haykin, *Adaptive Filter Theory, 4th Edition*, Prentice Hall, Information and System Sciences Series, 2001.