Generalized Self Spread-Spectrum Communications with Turbo Soft Despreading and Decoding

Stefano Tomasin and Daniele Veronesi

Abstract: **Self-spreading (SSP) is a spread spectrum technique where the spreading sequence is generated from data bits. Although SSP allows communications with low probability of interception by unintended receivers, despreading by the intended receiver is prone to error propagation. In this paper, we propose both a new transmitter and a new receiver based on SSP with the aim to a) reduce error propagation and b) increase the concealment of the transmission. We first describe a new technique for the generation of SSP spreading sequence, which generalizes SSPs of existing literature. We include also coding at the transmitter, in order to further reduce the effects of error propagation at the receiver. For the receiver, we propose a turbo architecture based on the exchange of information between a soft despreader and a soft-input soft-output decoder. We design the despreader in order to fully exploit the information provided by the decoder. Lastly, we propose a *chip decoder* that extracts the information on data bits contained in the spreading sequence from the received signal. The performance of the proposed scheme is evaluated and compared with existing spread-spectrum systems.**

Index Terms: **Low probability of interception, self spread spectrum, turbo despreader.**

I. INTRODUCTION

The use of spread spectrum (SS) techniques to increase security in wireless communications has been widely considered in the past for both military and civil applications [1], [2]. Together with cryptography systems, which ensure data secrecy, SS transmission reduces the probability of interception and enhances the resilience to both jamming and narrow-band interference.

The basic principle of SS is to modulate the data signal with a sequence that enlarges the original spectrum. The design of sequences—and more in general systems—that ensure a low probability of interception (LPI) has been a widely studied topic. Among the proposed techniques, we mention chaotic systems [3] and random spreading [4], which have found limited applications because of the difficulty in the implementation of the receiver. In code spreading, [5], [6] spreading and coding are substituted by a single low-rate code and a Viterbi decoder is used at the receiver instead of a traditional despreader. However, this solution is infeasible for a practical implementation when the spreading factor and the state of the code are large.

A reduction of complexity has been achieved by generating the spreading sequence from past transmitted data, as proposed in [7] and [8], with a technique that we denote self spread-spectrum (SSP). Although the dependency of the spreading sequence on the data enhances LPI, it also poses a detection problem for the intended user. In fact, the despreading sequence must be computed at the receiver from past detected bits and error propagation in the despreading process can severely degrade performance. In order to alleviate error propagation, differential encoding has been considered in [9], while the integration of SSP and coding has been investigated in [10], where only hard decoded bits are used for despreading.

In this paper, we consider a coded SSP transmission and we propose: a) An SSP transmission scheme that generalizes the original scheme of [9]; b) a receiver that performs decoding and despreading in a turbo fashion, and c) a technique to exploit the information on data bits carried by the spreading sequence *(chip decoding)*.

We consider an SSP transmission scheme where the spreading sequence is a general binary function of a block of past transmitted bits which constitute the spreading state. The general architecture provides a complete and flexible description of SSP systems. Moreover, by decoupling the size of the spreading state from the spreading factor, a large number of bits can be used for the generation of the spreading sequence without further enlarging the spectrum. As a result, for the same given spectrum, the generalized SSP provides more randomness in the spreading sequence then the original SSP system, and thus a lower probability of interception. At the same time, the dependency of the spreading sequence on the data can be exploited by the intended receiver to increase the reliability of data detection. Moreover, in order to improve the transmission reliability data are encoded with an error protection code before spreading.

As a receiver for the coded SSP transmission, we propose a turbo scheme that exchanges soft information between the despreader and the decoder. In particular, the despreading sequence is designed from the log likelihood ratio (llr) of the past despread bits with the aim of minimizing the mean square error (MSE) at the despreader output.

Lastly, with chip decoding the information conveyed by the spreading sequence about past transmitted bits is retrieved from the received chips at each detection process. To this end, once despreading has been performed, the soft information about the spread symbol is used to obtain new llr's on the bits that constitute the spreading sequence. This chip decoding allows to increase the reliability of the data estimate.

The paper is organized as follows. The system model of the coded SSP transmitter and the generation of the spreading sequence is described in Section II. In this section we also provide a description of the conventional despreading receiver that will be considered for performance evaluation. Section III describes the turbo SSP receiver and the details on the exchange

Manuscript received June 9, 2005; approved for publication by Marco Lops, Division I Editor, June 8, 2006.

The authors are with the Department of Information Engineering, University of Padova, via Gradenigo 6/B, Padova, Italy 35131, email: {tomasin, daniele}@dei.unipd.it.

This paper has been presented in part at the VTC 2004 spring and VTC 2004 fall.

1229-2370/06/$10.00 © 2006 KICS

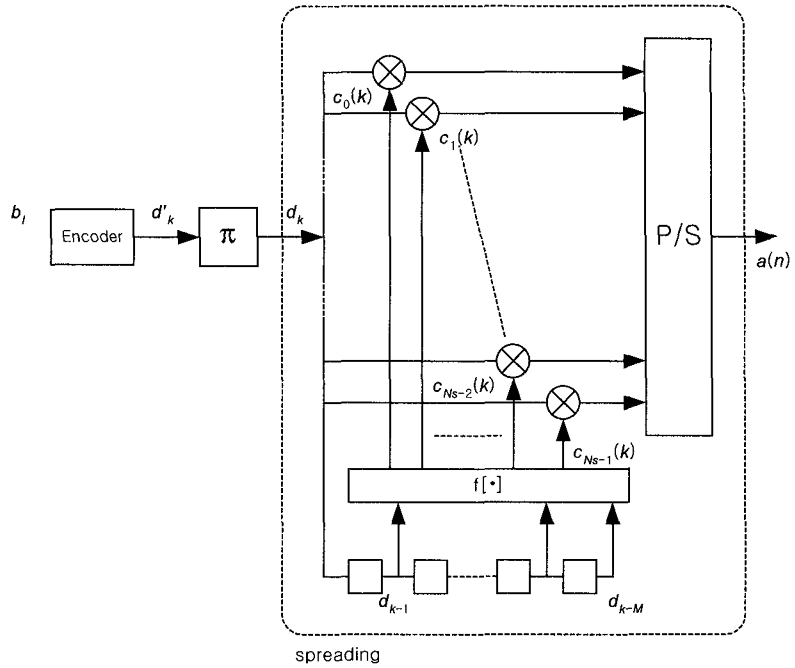

Fig. 1. SSP transmitter with coding.

of information among blocks. The soft despreader used in the turbo receiver is described in Section IV, where we also provide a description of the chip decoder. Numerical results on the performance of the proposed scheme are provided in Section V. Conclusions are outlined in Section VI.

II. SYSTEM MODEL

We consider a coded spread spectrum transmitter, according to the scheme of Fig. 1. A source provides a stream of bits which will be encoded by a convolutional encoder. In particular, the encoder processes blocks of K_b bits and provides blocks of

$$K_d = K_b/R + K_0 \tag{1}$$

bits, where R is the coding rate and K_0 is due to trellis termination. Let us indicate with[1]

$$\mathbf{d}' = [d'(0), d'(1), \cdots, d'(K_d - 1)] \tag{2}$$

the block of encoded bits ($d'(k) \in \{-1, +1\}$), which is interleaved into the block $\mathbf{d}$ before spreading. According to the scheme of Fig. 1, the interleaving process is indicated as

$$\mathbf{d} = \pi[\mathbf{d}']. \tag{3}$$

Lastly, spreading follows, which enlarges the spectrum through the sequence of N_S symbols $c_l(k)$, $l = 0, 1, \cdots, N_S - 1$, obtained from past transmitted bits and provides the signal at chip time

$$\begin{aligned} a(kN_S + l) &= c_l(k)d(k) \\ k &= 0, 1, \cdots, K_d - 1 \\ l &= 0, 1, \cdots, N_S - 1\,. \end{aligned} \tag{4}$$

[1] We use boldface for row vectors, e.g., $\mathbf{x}$, while element l of $\mathbf{x}$ is denoted with x_l.

According to the choice of $\{c_l(k)\}$, different SS systems are obtained. For example, in direct sequence (DS) SS systems with long spreading sequences, $\{c_l(k)\}$ are delayed versions of a common sequence of period $2^M - 1$, generated by a shift register with feedback having M memory blocks. In an SSP system, $\{c_l(k)\}$ is obtained from data bits that have been previously transmitted. Details on the generation of the SSP spreading sequence are provided in Section II-A.

As transmission scenario, we consider narrowband communications, and in particular we focus on AWGN communications. The received signal is then modeled as follows

$$r(n) = a(n) + w(n) \tag{5}$$

where $w(n)$ is the additive white Gaussian noise term with zero mean and variance σ_w^2.

A. Spreading Sequence Generation in SSP

The spreading sequence of an SSP system is generated from the M past transmitted bits, that are stored into the *spreading state* vector $\mathbf{D}(k)$. Initially, we set

$$\mathbf{D}(0) = \mathbf{p} \tag{6}$$

and for $k > 0$

$$D_m(k) = \begin{cases} d(k-1) & m = 0 \\ D_{m-1}(k-1) & m = 1, \cdots, M-1 \end{cases}$$

where $\mathbf{p}$ is the *starting state* known at the intended receiver. Obviously, the starting state should not be known to an unintended receiver, in order to ensure LPI.

The vector of the spreading sequence $\mathbf{c}(k)$ associated with the input bit $d(k)$ is a function of the state vector $\mathbf{D}(k)$, and of the time k, i.e.,

$$\mathbf{c}(k) = [c_0(k), c_1(k), \cdots, c_{N_S-1}(k)] = \mathbf{f}[\mathbf{D}(k), k]\,. \tag{7}$$

Note that the mapping function $\mathbf{f}$ can be time-varying to improve LPI.

As an example of mapping function, we consider the product of elements of the state $\mathbf{D}(n)$ and we denote the resulting scheme as *product SSP* (P-SSP). For example, the mapping function of a P-SSP system with $M = 2N_S$ is

$$c_l(k) = f_l[\mathbf{D}(k)] = D_l(k) \cdot D_{2N_S-l-1}(k) \quad (8)$$
$$l = 0, 1, \cdots, N_S - 1 \quad (9)$$

and from (7), we obtain

$$f_l[\mathbf{D}(k)] = d(k-l-1)d(k-2N_S+l) \quad (10)$$

for $k > 0$. Note that the newest bits are multiplied with the oldest ones. This arrangement will ensure a better protection against error propagation at the receiver.

In general, the binary mapping function (10) can be extended for states of length M, with $M = IN_S$, where $I > 0$ is an integer. By defining the permutation function

$$\gamma(m,l) = \begin{cases} l + mN_S & \text{if } m \text{ is even} \\ (m+1)N_S - l - 1 & \text{if } m \text{ is odd} \end{cases} \quad (11)$$

the l-th element of the mapping function for a general P-SSP system is

$$f_l[\mathbf{D}(k)] = \prod_{m=0}^{M/N_S-1} D_{\gamma(m,l)}(k), \quad l = 0, 1, \cdots, N_S - 1. \quad (12)$$

For $M = N_S$, the spreading sequence coincides with the state and

$$c_l(k) = D_l(k) = d(k-1-l) \quad (13)$$

is the particular SSP sequence of [9].

In order to despread the SSP signal at the receiver, we consider two solutions. In Section II-B, we present a despreading receiver where past detected data generate the despreading sequence. With this solution, the coding is not exploited in the despreading process. Moreover, the generation of the despreading sequence from hard detection of the despread symbol enhances error propagation. In Section III, instead we propose a receiver that implements soft despreading and exploits coding in the despreading process through a turbo structure that exchanges information between decoding and despreading. Moreover, we include in the receiver a *chip decoder* that provides additional information on the bits used for spreading.

B. Despreading SSP Receiver

The receiver for a conventional SS signal includes a despreader followed by a detector. A similar scheme can be used also for SSP systems, with the difference that the despreading sequence is generated from past detected data. The despreading SSP receiver for (13) introduced in [9] is here extended for a general SSP transmission.

The terminal stores a state vector that is initialized with the *starting state* known to the intended user, i.e.,

$$\hat{\mathbf{D}}(0) = \mathbf{p} \quad (14)$$

and it is updated with the detected bits.

For each block of N_S samples after the matched filter

$$\mathbf{r}(k) = [r(kN_S), r(kN_S+1), \cdots, r(kN_S+N_S-1)] \quad (15)$$

the receiver performs the following operations.

1. Using the mapping function $\mathbf{f}(\cdot)$, the despreading sequence is generated as
$$\hat{\mathbf{c}}(k) = \mathbf{f}[\hat{\mathbf{D}}(k), k]. \quad (16)$$
The generation of the spreading sequence from past detected bits $\hat{\mathbf{D}}(k)$ is the main novelty when compared to the DS-SS despreader, where the sequence is fixed for each user.
2. Despreading is performed with the estimated spreading sequence $\hat{\mathbf{c}}(k)$ to yield
$$z(k) = \frac{1}{\theta} \sum_{l=0}^{N_S-1} \hat{c}_l(k) r_l(k) \quad (17)$$
where $\theta = \sum_{l=0}^{N_S-1} |\hat{c}_l(k)|^2$ is a normalization factor. The despread sample is passed to the de-interleaver and decoder for further processing.
3. A threshold detector (denoted as $Q[\cdot]$) yields
$$\hat{d}(k) = Q[z(k)]. \quad (18)$$
4. The state vector $\hat{\mathbf{D}}(k+1)$ is obtained from $\hat{\mathbf{D}}(k)$ by inserting the newly detected symbol into the shifted state, and removing the oldest bit $\hat{D}_{M-1}(k)$, i.e.,
$$\begin{aligned} \hat{D}_m(k+1) &= \hat{D}_{m-1}(k), \quad m = 1, 2, \cdots, M-1 \\ \hat{D}_0(k+1) &= \hat{d}(k). \end{aligned} \quad (19)$$

Note that from (16) and (17), errors in past-detected bits corrupt the generation of the despreading sequence and consequently despreading and detection of the next symbols. In order to reduce the error propagation, differential spreading has been proposed in [9], which however yields a performance penalty for a high signal to noise plus interference ratio (SNIR). Moreover, both coherent and differential schemes based on despreading do not exploit either the coding or the dependency on data of the spreading sequence in order to improve the reliability of estimates.

III. TURBO SSP RECEIVER

The turbo receiver for coded SSP transmissions performs iteratively decoding and despreading of the signal and exchanges soft information on the bits between the two processes. The basic idea of the turbo SSP receiver is to exploit the error correction capabilities of the code for the generation of the despreading sequence.

The scheme of the turbo receiver is shown in Fig. 2. Each stage of the turbo process is implemented by five blocks. The first two blocks generate the log likelihood ratios (llr's) on the bits that will be used for decoding: a) A soft despreader reverts the SSP of the transmitter by using the a-priori information provided by the previous turbo iteration and b) a signal to probability (StP) block derives the extrinsic llr of the transmitted bits.

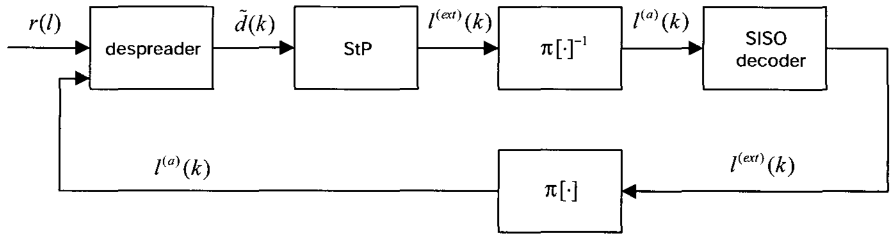

Fig. 2. Turbo SSP receiver.

The decoding process is implemented into three stages: i) A de-interleaver $\pi[\cdot]^{-1}$ reverses the interleaving operation of the transmitter and feeds ii) a soft-input soft-output (SISO) decoder; lastly iii) the soft symbols provided by the decoder are interleaved by the interleaver $\pi[\cdot]$ and passed to the soft despreader, as a-priori information.

A. Soft Despreading

A distinguishing feature of the turbo receiver with respect to the despreading receiver is the soft despreader that uses soft information from the previous iteration/despreding.

The N_S chips of each received block signal $\mathbf{r}(k)$ are combined through the despreading sequence having soft (not binary) values

$$\boldsymbol{\alpha}(k) = [\alpha_0(k), \alpha_1(k), \cdots, \alpha_{N_S-1}(k)] \tag{20}$$

to yield the despread sample

$$\tilde{d}(k) = \boldsymbol{\alpha}(k)\mathbf{r}(k)^T. \tag{21}$$

We provide details for the design of the despreading sequence $\boldsymbol{\alpha}(k)$ in Section IV where we consider as design criterion the minimization of the mean square error (MSE) at the despreader output. We will also consider a more elaborate despreading architecture which includes chip decoding, i.e., the retrieval of additional information from the spreading chips on the data bits.

B. Signal to Probability Block

Soft despreading provides an estimate of the transmitted symbol $\tilde{d}(k)$ which is used by the StP block to obtain the llr that will be used by the convolutional decoder. The llr provided at time k by the StP block must not depend on a-priori information on the symbols. On the other hand, the soft despreader uses the complete information (both apriori and extrinsic) on the past received bits and it provides complete llr through $\tilde{d}(k)$. In order to remove the a-priori information from the llr computed on $\tilde{d}(k)$, we first compute the llr on the despread sample, indicated as $\ell(k)$, and then subtract the a-priori llr provided by SISO decoder at the previous iteration

$$\ell^{(a)}(k) = \ln\left(\frac{P[d(k)=+1]}{P[d(k)=-1]}\right) \tag{22}$$

and the extrinsic llr of the despread bit can be written as

$$\ell^{(ext)}(k) = \ell(k) - \ell^{(a)}(k). \tag{23}$$

At the first iteration no information is available on the data bits except for the starting state and

$$\ell^{(a)}(k) = \begin{cases} p_k L & k = 0, 1, \cdots, M-1 \\ 0 & \text{otherwise} \end{cases} \tag{24}$$

where L is a large real number (ideally $L = \infty$) and p_k is the k-th element of $\mathbf{p}$.

On the other hand, by indicating with ρ the sample after soft despreading, the llr after the despreader is defined as

$$\ell(k) \triangleq \ln\left(\frac{P[d(k)=1|\tilde{d}(k)=\rho]}{P[d(k)=-1|\tilde{d}(k)=\rho]}\right). \tag{25}$$

By the Bayes rule, we can write

$$\begin{aligned}\ell(k) &= \ln\left(\frac{P[\tilde{d}(k)=\rho|d(k)=+1]}{P[\tilde{d}(k)=\rho|d(k)=-1]}\right) \\ &\quad + \ln\left(\frac{P[d(k)=+1]}{P[d(k)=-1]}\right).\end{aligned} \tag{26}$$

Hence, from (22) and (23), the extrinsic llr can be written as

$$\ell^{(ext)}(k) = \ln\left(\frac{P[\tilde{d}(k)=\rho|d(k)=+1]}{P[\tilde{d}(k)=\rho|d(k)=-1]}\right). \tag{27}$$

For the computation of the extrinsic llr we assume that sample $\tilde{d}(k)$ can be modeled as the output of the transmission of the spread symbol on an AWGN channel, i.e., from (21)

$$\tilde{d}(k) = \beta(k)d(k) + \tilde{w}(k) \tag{28}$$

where $\beta(k)$ is the gain of the user signal after soft despreading and $\tilde{w}(k)$ is the noise term. From (21), we obtain the useful signal signal gain as

$$\beta(k) = \boldsymbol{\alpha}(k)\mathbf{c}(k)^T \tag{29}$$

and the noise term after soft despreading turns out to be

$$\tilde{w}(k) = \sum_{l=0}^{N_S-1} \alpha_l(k) w(kN_S + l) \tag{30}$$

which can be described as a Gaussian random variable having zero mean and variance

$$\sigma_{\tilde{w}}^2 = \sigma_w^2 \sum_{l=0}^{N_S-1} |\alpha_l(k)|^2. \tag{31}$$

From the AWGN model, the probability of having received ρ, given that the symbol ξ has been transmitted is easily computed as

$$P[\tilde{d}(k) = \rho | d(k) = \xi] = \frac{1}{\sqrt{2\pi}\sigma_{\tilde{w}}} e^{-\frac{|\rho - \beta(k)\xi|^2}{2\sigma_{\tilde{w}}}} \quad (32)$$

where for a BPSK transmission $\xi \in \{-1, 1\}$.

Lastly, by substituting the probability (32) in (27), we obtain the extrinsic llr as

$$\ell^{(ext)}(k) = \frac{2\rho\beta(k)}{\sigma^2_{\tilde{w}}} \quad (33)$$

which is the value provided by the StP block.

C. SISO Decoder

The SISO decoder is based on a symbol-wise maximum a posteriori probability (MAP) criterion and it is implemented with the BCJR algorithm [11]. This block generates llr on the transmitted bits by exploiting the error correction capabilities of the code.

At each iteration $\ell^{(a)}(k)$ are updated according to the extrinsic probability provided by the soft despreader.

At the last iteration, the llr are used to obtain the best estimate of the transmitted data bits, to be delivered to higher layer services.

IV. DESPREADING SEQUENCE DESIGN

In the soft despreading procedure described in Section III-A, the despreading sequence is obtained from previously transmitted bits and the despreader will be affected by the reliability of past decisions. In order to mitigate the effects of error propagation, we propose to use as despreading sequence $\{\alpha_l(k)\}$ a set of soft values instead of the hard detected bits. In particular, the despreading sequence is designed with the aim of minimizing the mean square error of the de-spread sample at the detector input

$$J(k) = \mathrm{E}[|\tilde{d}(k) - d(k)|^2] \quad (34)$$

where E[·] denotes expectation. From the AWGN model (28), $J(k)$ can be rewritten as

$$J(k) = \mathrm{E}\left[\left|d(k)\left(\beta(k) - 1\right) + \sum_{l=0}^{N_S-1} \alpha_l(k) w(kN_S + l)\right|^2\right]. \quad (35)$$

The MSE is related to the reliability of the estimates of the bits which are used in the spreading sequence. This reliability is reflected in the average and the variance of the chip $c_l(k)$ evaluated with respect to the a-priori information on the bits used for the generation of the chip. We indicate with $\bar{c}_l(k)$ the average of the chip $c_l(k)$, and we define the correlation matrix of the spreading sequence

$$\mathbf{R_c}(k) = \mathrm{E}[\mathbf{c}(k)^* \mathbf{c}(k)^T] \quad (36)$$

$$[\mathbf{R_c}(k)]_{p,q} = \begin{cases} \mathrm{E}[|c_p(k)|^2] & \text{if } p = q \\ \mathrm{E}[c_p(k)]\mathrm{E}[c_q(k)] & \text{elsewhere} \end{cases} \quad (37)$$

where * denotes the complex conjugate and the expectation is taken with respect to the a-priori probability of the bits that generate the spreading sequence, as from receiver's knowledge. Lastly, let us indicate the power of the data symbols as

$$\mathtt{M}_d = \mathrm{E}[|d(k)|^2]. \quad (38)$$

Under the assumption that the transmitted symbols and noise samples are uncorrelated, the MSE can be rewritten as

$$\begin{aligned} J(k) &= \mathtt{M}_d\Big[\boldsymbol{\alpha}(k)^*\Big(\mathbf{R_c}(k) + \frac{\sigma_w^2}{\mathtt{M}_d}\Big)\boldsymbol{\alpha}(k)^T + \\ &\quad -2\mathrm{Re}[\bar{\mathbf{c}}(k)\boldsymbol{\alpha}(k)^T] + 1\Big]. \end{aligned} \quad (39)$$

As despreading sequence, we select the set $\boldsymbol{\alpha}(k)$ that minimizes the MSE, given a certain knowledge of the average and correlation of chips. Thus, by setting to zero the gradient of $J(k)$ with respect to $\boldsymbol{\alpha}(k)$, we get

$$\boldsymbol{\alpha}(k) = \Big(\mathbf{R_c}(k) + \frac{\sigma_w^2}{\mathtt{M}_d}\Big)^{-1} \bar{\mathbf{c}}(k)^{*T}. \quad (40)$$

Once the average values of the chip is available, we can also derive the gain of the useful signal $\beta(k)$ in the AWGN model. In particular, we consider the following estimate

$$\overline{\beta}(k) = \boldsymbol{\alpha}(k)\bar{\mathbf{c}}(k)^T. \quad (41)$$

In the following sections, we propose two techniques for the computation of the average spreading sequence, $\bar{\mathbf{c}}(k)$. In the first case, the average is computed from the a-priori information provided by the decoder and past despreading. As a refinement of this technique, we propose also to exploit the information contained in the spreading sequence.

A. Statistics Derivation

For the computation of the average of the chips to be used in the despreader, we propose to use the available llr on the bits that contribute to the spreading sequence. The llr comes from the despreading process of previous chips, as well as from the decoding of previous turbo iterations. In particular, an estimate of the bit $d(k)$ is obtained (see [13]) as

$$\overline{d}(k) = \mathrm{E}[d(k)] = \tanh\left(\frac{\ell^{(a)}(k) + \ell^{(ext)}(k)}{2}\right). \quad (42)$$

The average and power of the chip sequence $\mathrm{E}[c_l(k)]$ and $\mathrm{E}[|c_l(k)|^2]$ are derived from (7) by writing the binary function $\mathbf{f}[\mathbf{D}(k),k]$ as a composition of sums and products of the estimates of the state bits. The AND operations in $\mathbf{f}[\mathbf{D}(k),k]$ are substituted with products of soft estimates, while OR operations are substituted with averages.

In particular, for P-SSP, from (12), we have

$$\bar{c}_l(k) = \mathrm{E}[c_l(k)] = \prod_{m=0}^{M/N_S-1} \overline{d}(k - 1 - l - mN_S) \quad (43)$$

$$\mathtt{M}_{c_l} = \mathrm{E}[|c_l(k)|^2] = 1. \quad (44)$$

B. Improved Statistics with Chip Decoding

The SSP signal carries information on data both in the spread symbol and in the spreading sequence. Indeed, the SSP process can be seen as a coding procedure that generates a coded and spread signal from the data. The optimum maximum likelihood (ML) decoder for SSP can be implemented with a Viterbi algorithm where the trellis is built on the spreading state. However, the computational complexity of the resulting scheme grows exponentially with the size of the state M, [12]. Considering that a better performance is achieved with longer states, a practical implementation of the ML decoder is infeasible. We resort instead to a suboptimal approach that generates an estimate of the spreading bits from the received chips and iterates between despreading and chip estimate.

The chip decoding is based on the observation that, once the received signal has been despread, an estimate of the data-dependent spreading sequence is obtained from the received block of N_S chips with the knowledge of the despread symbol.

Hence, the soft despreader with *chip decoding* comprises two steps: a) *Despreading* on blocks of N_S input samples, as described in the previous section; b) computation of a new llr of the state bits using the information on the despread symbol.

The new llr values computed in the step b) are used to further refine the data estimation. In general, by indicating with $\ell^{(cd)}(k,p)$ the new llr computed on symbol k at time pN_ST with $p > k$, the apriori information on symbol $d(k)$ becomes

$$\ell^{(a)'}(k) = \ell^{(a)}(k) + \sum_{q=1}^{p-k} \ell^{(cd)}(k,q). \tag{45}$$

Using this new value in (42), we obtain a more reliable estimated of $\overline{d}(k)$.

In order to provide a better insight into the chip decoder, we consider the P-SSP system.

In the general case of $M > N_S$, the llr of each state bit pertaining to the chip $c_l(k)$ is obtained by using both the llr of the spread bit and the llr of the other bits that contribute to $c_l(k)$. First, a MSE estimate of the state bit $D_l(k)$ is derived as

$$\tilde{D}_l^{(cd)}(k) = r_{(l \bmod N_S)}(k)\psi_l(k) \tag{46}$$

and from (12), we obtain

$$\begin{aligned}\tilde{D}_l^{(cd)}(k) &= D_l(k)d(k)\psi_l(k)\prod_{\substack{m=0\\m\neq l}}^{M-1} D_m(k) \\ &\quad + w(l+kN_S)\psi_l(k) \\ &= \beta_l'(k)D_l(k) + \tilde{w}'(l+kN_S)\end{aligned} \tag{47}$$

where $\psi_l(k)$ is the coefficient that minimizes the MSE of the chip estimate. By assuming that the other bits of the state have a statistical distribution provided by the available llrs, the minimization of the MSE

$$J_l'(k) = \mathrm{E}[|\tilde{D}_l^{(cd)}(k) - D_l(k)|^2] \tag{48}$$

provides

$$\psi_l(k) = \left(\mathrm{M}_d \prod_{\substack{m=0\\m\neq l}}^{M-1} \mathrm{E}[|D_m(k)|^2] + \frac{\sigma_w^2}{\mathrm{M}_d}\right)^{-1} \overline{d}(k)^* \prod_{\substack{m=0\\m\neq l}}^{M-1} \overline{D}_m(k)^*. \tag{49}$$

From (47) the variance of the noise w' is

$$\sigma_{\tilde{w}'}^2 = \mathrm{E}[|\tilde{w}'(k)|^2] = |\psi_l(k)|^2\sigma_w^2 \tag{50}$$

while the gain factor $\beta_l'(k)$ is approximated as

$$\beta_l'(k) \approx \overline{d}(k)\psi_l(k)\prod_{\substack{m=0\\m\neq l}}^{M-1} \overline{D}_m(k) \tag{51}$$

where

$$\overline{D}_q(k) = \mathrm{E}[D_q(k)] \tag{52}$$

is computed from a-priori information as $\overline{d}(k)$. Similarly to (33), the llr of $\tilde{D}_l^{(cd)}(k)$ can be written as

$$\ell^{(cd)}(k-1-l,k) = \frac{2\overline{d}(k)\psi_l(k)\tilde{D}_l^{(cd)}(k)\prod_{\substack{m=0\\m\neq l}}^{M-1}\overline{D}_m(k)}{|\psi_l(k)|^2\sigma_w^2}, \quad l = 0,1,\cdots,M-1. \tag{53}$$

In the special case $M = N_S$, the MSE coefficient is the same for all chips

$$\psi_l(k) = \left(\mathrm{M}_d + \frac{\sigma_w^2}{\mathrm{M}_d}\right)^{-1}\overline{d}(k)^* \tag{54}$$

while the new llr turns out to be

$$\ell^{(cd)}(k-l-1,k) = \frac{2\overline{d}(k)\tilde{D}_l^{(cd)}(k)}{|\psi_l(k)|^2\sigma_w^2}, \quad l = 0,1,\cdots,M-1. \tag{55}$$

V. NUMERICAL RESULTS

The performance of the proposed turbo receiver has been evaluated for an AWGN transmission. The function $\mathbf{f}[\mathbf{D}(k),k]$ has been chosen according to the P-SSP scheme with $N_S = 16$. The data bits are grouped into frames of 510 bits, each of which is encoded with the recursive systematic convolutional code having rate $1/2$ and generating polynomials $[133_8, 171_8]$. Before spreading, the encoded bits are interleaved by an S-random interleaver [14] with $S = round(.5 * \sqrt{.5 * N})$, where $N = 1024$. Performance is evaluated in terms of both bit error rate (BER) and frame error rate (FER). For comparison purposes, we evaluate also the performance of a conventional direct sequence spread-spectrum (DS-SS) system, where a fixed spreading sequence is used. The turbo SSP receiver is denoted as TSSP and when chip decoding is used, the scheme is denoted TSSP-CD.

Fig. 3 shows the FER for TSSP and TSSP-CD systems as a function of the average E_b/N_0 for various iterations. In this case

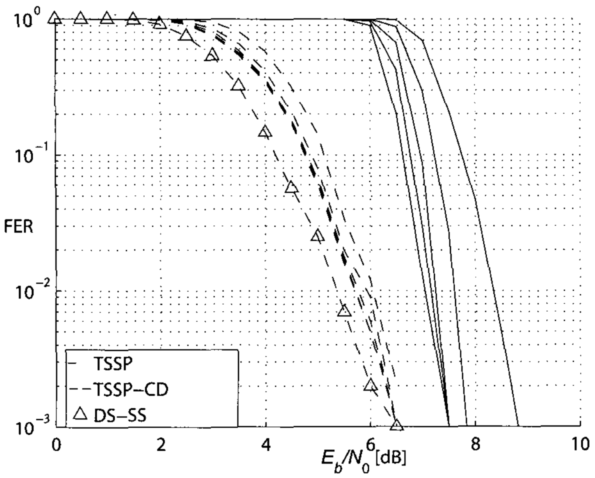

Fig. 3. Average FER for TSSP (solid lines) and TSSP-CD (dashed lines) for iterations from 1 to 5. $M = N_S = 16$. The dashed line with △ shows the FER of DS-SS system.

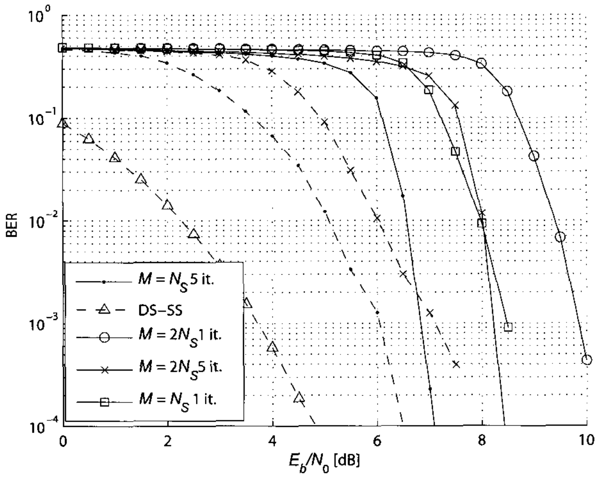

Fig. 4. Average BER for various spread spectrum systems. Solid lines: TSSP, dashed lines: TSSP-CD.

$M = N_S$. From the figure, we observe that TSSP is able to reduce only partially the error propagation phenomenon, while the integration of the chip decoding provides a significant improvement of performance. We also note that the iterative process is more beneficial for TSSP, where the initial estimate is less accurate than for TSSP-CD.

Fig. 5 shows the FER for TSSP and TSSP-CD where $M = 2N_S$. In this case, a longer state worsen the performance with respect to the case $M = N_S$, while ensuring higher concealment capabilities. Moreover, we observe that the CD scheme is able to benefit from the additional data-dependency of the spreading sequence thus providing better performance.

Fig. 4 shows the BER of the systems for the two choices of M. Also in this case we obtain similar conclusions to those of the FER performance. In particular, TSS-CD provides a significant advantage for the $M = 2N_S$ scheme with respect to the case with no CD and no turbo receiver.

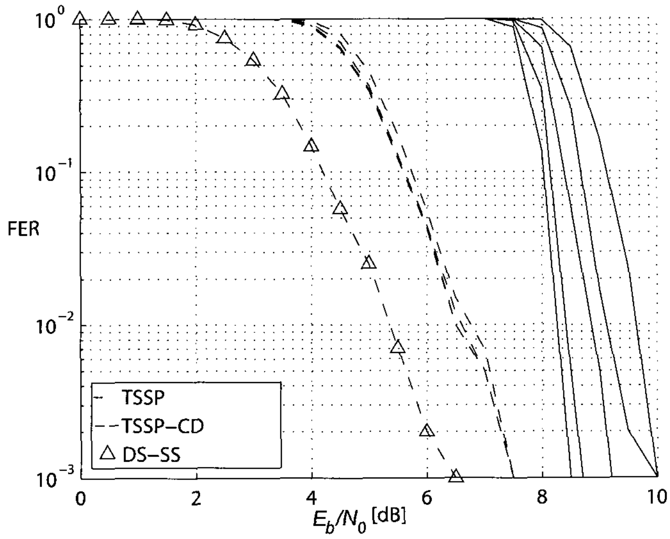

Fig. 5. Average FER for TSSP (solid lines) and TSSP-CD (dashed lines) for iterations from 1 to 5. $M = 2N_S = 32$. The dashed line with △ shows the FER of DS-SS system.

VI. CONCLUSION

In this paper, we proposed a transmit and receive technique based on the self spread-spectrum communications. The general framework of SSP transmission allows to design a variety of spreading functions that may achieve different concealment and coding capabilities. For the receiver, we proposed a turbo architecture that exchanges information between the soft despreader and the SISO decoder. Moreover, the use of chip decoding improves the data estimate by exploiting the information on data contained in the spreading sequence. Performance evaluations show that the turbo receiver, together with the chip decoder, is effective in decoding the data.

ACKNOWLEDGMENTS

This work has been partially supported by the Italian ministry for University (MIUR), with FIRB projects.

REFERENCES

[1] R. Scholtz, "The origins of spread-spectrum communications," *IEEE Trans. Commun.*, vol. 30, pp. 822–852, May 1982.

[2] "Part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications," IEEE Standard Specifications, 1999.

[3] M. Sushchik, L. S. Tsimring, and A. R. Volkovskii, "Performance analysis of correlation-based communication schemes utilizing chaos," *IEEE Trans. Circuits Syst. I*, vol. 47, no. 12, pp. 1684–1691, Dec. 2000.

[4] S. Verdu and S. Shamai, "Spectral efficiency of CDMA with random spreading," *IEEE Trans. Inform. Theory*, vol. 45, no. 2, pp. 622–640, Mar. 1999.

[5] A. J. Viterbi, "Very low rate convolutional codes for maximum theoretical performance of spread-spectrum multiple-access channels," *IEEE J. Select. Areas Commun.*, vol. 8, no. 4, pp. 641–649, May 1990.

[6] P. Frenger, P. Orten, and T. Ottoson, "Code-spread CDMA using low-rate convolutional codes," in *Proc. IEEE ISSSTA'98*, 1998, pp. 374–378 .

[7] L. Nguyen, "Self-encoded spread spectrum communications," in *Proc. IEEE MILCOM'99*, vol. 1, 1999, pp. 182–186.

[8] L. Nguyen, "Self-encoded spread spectrum and multiple access communications," in *Proc. IEEE ISSSTA 2000*, vol. 2, Sept. 2000, pp. 394–398.

[9] Y. Kong, L. Nguyen, and W. M. Jang, "Self-encoded spread spectrum modulation with differential encoding," in *Proc. IEEE 7-th ISSSTA 2002*, vol. 2, 2002, pp. 471–474.
[10] W. M. Jang, L. Nguyen, and M. Hempel, "Self-encoded spread spectrum and turbo coding," *J. Commun. and Networks*, vol. 6, pp. 6–18, Mar. 2004.
[11] L. R. Bahl, J. Cocke, F. Jelinek, and R. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inform. Theory*, vol. 20, pp. 284–284, Mar. 1974.
[12] N. Benvenuto and G. Cherubini, *Algorithms for Communications Systems and Their Applications*, Chichester: Wiley, 2002.
[13] M. Tuchler, R. Koetter, and A. Singer, "Turbo equalization: Principles and new results," *IEEE Trans. Commun.*, vol. 50, no. 5, pp. 754–767, May 2002.
[14] C. Heegard and S. Wicker, *Turbo Coding*, Kluwer Academic Publishing, 1999.

Stefano Tomasin is assistant professor at University of Padova, Italy. After the Laurea degree in Telecommunications Engineering (1999), he was with the IBM Research Laboratory, Zurich, Switzerland, working on signal processing for magnetic recording systems. In 2000, he started the Ph.D. course in Telecommunication Engineering, working on multicarrier communication systems for wireless broadband communications. In the academic year 2001–2002, he was with Philips Research, Eindhoven, the Netherlands, studying multicarrier transmission for mobile applications. After receiving the Ph.D. degree (2002), he joined University of Padova first as contractor researcher for a national research project and then as assistant professor. In the second half of 2004, he was visiting faculty at Qualcomm, San Diego (CA).

Daniele Veronesi was born in Rovereto, Italy, on August 4, 1979. He received the Laurea degree in Telecomunication Engineering (2003) from the University of Padova, Italy. From January 2004, he is a Ph.D. student at the University of Padova. During the academic year 2004–2005, he has been six months at the University of Massachusetts, Amherst (USA) as exchanging student. His major interests are power control in cellular system, signaling in cellular system, and cooperative systems.