in "Systems and Control: Theory and Applications", published by WSES, 2000, pp. 257-262, ISBN 960-8052-11-4

Spread Spectrum Codes Identification by Neural Networks

Céline BOUDER, Gilles BUREL
Laboratoire d'Electronique et Systèmes de Télécommunications (UMR CNRS 6616)
Université de Bretagne Occidentale
6 avenue Le Gorgeu, BP 809
29285 BREST cedex, FRANCE
Celine.Bouder@univ-brest.fr, Gilles.Burel@univ-brest.fr

Abstract: - In the context of spectrum surveillance, a method to recover the code of direct sequence spread spectrum signal is presented, whereas the receiver has no knowledge of the transmitter's spreading sequence. The approach is based on an artificial neural network which is forced to model the received signal. Experimental results show that the method provides a good estimation, even when the signal power is below the noise power.

Keywords: - Spread Spectrum Communications, Spectrum Surveillance, Identification, Pseudo-Random sequences, Artificial Neural Networks.

1 Introduction

Although spread spectrum communications were initially developed for military applications, they are now widely used for commercial ones, especially for code division multiple access (CDMA), or global positioning systems (GPS) [1]. They are mainly used to transmit at low power without interference due to jamming, to others users or to multipath propagation. The spread spectrum techniques are useful for secure transmissions, because the receiver has to know the sequence used by the transmitter to recover the transmitted data, using a correlator [2, 3, 4].

Our purpose is to automatically determine the spreading sequence, whereas the receiver has no knowledge of the transmitter's pseudo-noise (PN) code.

In the next section, we present the technique of direct sequence spread spectrum (DS-SS) and we explain the difficulty to recover the data in an unfriendly context. Then, we introduce our method, which uses artificial neural networks to solve the problem. Finally, section 4 shows experimental results in various configurations.

2 DS-SS technique

In order to spread the signal power over a broadband channel, far in excess of the minimum bandwidth necessary to transmit the data, the direct sequence spread spectrum (DS-SS) technique consists in multiplying the information signal with a periodic pseudo-noise sequence.

2.1 A simple model

Let us note b(t) the information bearing signal

$$b(t) = \sum_{-\infty}^{+\infty} b_n p(t - nT_b) \qquad (1)$$

where $b_n = \pm 1$ with equal probability and $p(t)$ is a rectangular pulse of duration T_b.

Let us note y, the PN sequence of length P :

$$y = y_0, y_1, \cdots, y_{P-1} \qquad (2)$$

The transmitted signal $\hat{y}_n$ is the product of both waveforms. Let us consider a direct sequence spread spectrum system without noise :

$$\hat{y}_n = b_n y \qquad (3)$$

We assume the receiver knows this sequence and can despread the signal using a correlator :

$$\langle \hat{y}_n, y \rangle = \langle b_n y, y \rangle = b_n \langle y, y \rangle = b_n P \qquad (4)$$

according to the properties of PN sequences [5], the data information is then recovered.

However it becomes more challenging when the receiver does not know exactly the code used by the transmitter.

Let us note $\tilde{y}$ a sequence similar to y, but not exactly the same. Then using a correlator with $\tilde{y}$, we get :

$$\langle \hat{y}_n, \tilde{y} \rangle = \langle b_n y, \tilde{y} \rangle = b_n \langle y, \tilde{y} \rangle \qquad (5)$$

according to the properties of PN sequences, $\langle y, \tilde{y} \rangle$ is low [5] and then we do not recover the data information.

2.2 A realistic model

Typically direct sequence spread spectrum systems use binary or quadrature phase shift keying (BPSK or QPSK) data modulation. Usually the PN sequence is a binary maximal length sequence or a Gold sequence [4]. Sometimes complex signature sequences are used. It has been shown [6], that using complex codes provides an improvement of 3 dB (in comparison with binary Gold sequences) against users interference.

Here we consider a PSK data modulation, spread by a complex signature sequence. The baseband receiver signal at the output of the receiving filter can be written as :

$$s(t) = \sum_{k=-\infty}^{+\infty} a_k h(t - kT_s) + n(t) \qquad (6)$$

where $h(t)$ is the combined impulse response of the channel and the spreading code :

$$h(t) = \sum_{m=0}^{P-1} c_m p(t - mT_c) \qquad (7)$$

and $p(t) = (e * g * c)(t)$ (8)

P is the length of the spreading sequence.

$\{c_m, m = 0 \cdots P-1\}$ is the spreading sequence.

a_k is the symbol number k.

T_c is the chip period.

T_s is the symbol period $(T_s = PT_c)$.

$c(t)$ is the channel filter (that modelises the channel echoes).

$e(t)$ is the transmitting filter.

$g(t)$ is the receiving filter.

$n(t)$ is the noise at the output of the receiving filter.

The baseband channel noise is assumed to be white, gaussian and centered.

An interesting method to estimate $h(t)$ is proposed in [7]. It takes profit of blind identification techniques available for multiple FIR channels. Good results were obtained. The method implicitly assume that each symbol a_k has been precisely located in time. This is a strong requirement, since no method is known to perform time localization of the symbols without knowing the sequence. In this paper, we propose an approach that does not require knowledge of symbols times. It only needs previous estimate of the symbol period. The method is based on artificial neural networks techniques.

3 Estimation of the spreading sequence

To recover data information, we have to estimate $h(t)$, without knowing the transmitter's PN sequence. In this section we explain our method, which is based on artificial neural networks.

3.1 Theoretical analysis

The transmitted signal is the same as previously defined.

The symbol period T_s is assumed to be known, it can be estimated using cyclostationarity analysis [7]. The received signal is sampled, and we will note T_e the sampling period. We assume that T_e is such that $T_s = MT_e$ where M is an integer.

Let us note $\vec{s}(t)$ the M-dimensional vector below :

$$\vec{s}(t) = \left[s(t), s(t+T_e), \cdots, s(t+T_s-T_e) \right] \qquad (9)$$

where $\vec{h}(t)$ and $\vec{n}(t)$ are defined in the same way.

From the signal samples, we can create a matrix S with M rows and N columns, where N is the number of temporal windows of duration T_s in the signal used for estimation :

$$S = \begin{bmatrix} \vdots & \vdots & \cdots & \vdots \\ \vec{s}(t) & \vec{s}(t+T_s) & \cdots & \vec{s}(t+(N-1)T_s) \\ \vdots & \vdots & \cdots & \vdots \end{bmatrix} \qquad (10)$$

Let us note $t = mT_s + t_0$, where $0 \le t_0 < T_s$

From equation (6) we can write :

$$\vec{s}(t) = \sum_{k=-\infty}^{+\infty} a_k \vec{h}(t_0 + (m-k)T_s) + \vec{n}(t) \qquad (11)$$

$$\vec{s}(t) = \sum_{k=-\infty}^{+\infty} a_{m-k} \vec{h}(t_0 + kT_s) + \vec{n}(t) \qquad (12)$$

Let us note $\vec{h}_k(t_0)$ the vector below :

$$\vec{h}_k(t_0) = [h(t_0 + kT_s), \cdots, h(t_0 + (k+1)T_s - T_e)]^T$$

Hence we can write :

$$\vec{s}(t) = \sum_k a_{m-k} \vec{h}_k(t_0) + \vec{n}(t) \qquad (13)$$

Since the time extension of $h(t)$ is limited, the sum has been limited to values of k for which $\vec{h}_k(t_0)$ is not null. In the sequel, we assume for clarity, that

$h(t) \approx 0$ for t outside the interval $[0, T_s]$. Hence, $\vec{s}(t)$ can be written as follow :

$$\vec{s}(t) = a_m \vec{h}_0(t_0) + a_{m+1} \vec{h}_{-1}(t_0) + \vec{n}(t) \quad (14)$$

where $\vec{h}_0(t_0)$ is the M-dimensional vector containing the end of the spreading waveform (for a duration $T_s - t_0$) followed by zeros (for duration t_0) .

$\vec{h}_0(t_0) = [h(t_0), h(t_0 + T_e), \cdots, h(T_s - T_e), 0, \cdots, 0]^T$.

$\vec{h}_{-1}(t_0)$ is the M-dimensional vector containing zeros (for a duration $T_s - t_0$) followed by the beginning of the spreading waveform (for duration y .

$\vec{h}_{-1}(t_0) = [0, \cdots, 0, h(0), h(T_e), \cdots, h(t_0 - T_e)]^T$.

Hence, we can write the matrix S as follow :

$$S = \vec{h}_0 . \vec{a}_0^T + \vec{h}_{-1} . \vec{a}_1^T + \vec{n}(t) \quad (15)$$

where $\vec{a}_m = [a_m, a_{m+1}, \cdots, a_{m+N-1}]^T$

$\vec{h}_0(t_0)$ and $\vec{h}_{-1}(t_0)$ are orthogonal, and the noise is uncorrelated with the signal. Hence the subspace spanned by $\vec{h}_0(t_0)$ and $\vec{h}_{-1}(t_0)$ can be identified by a three layers neural network, whose hidden layer includes two neurons [8, 9, 10]. In fact we estimate $h(t)$ thanks to the second layer of weights.

3.2 Description of the artificial neural network

We create a feedforward network with three layers : a layer of the inputs, a hidden layer of two sigmoid neurons with hyperbolic tangent nonlinearities and an output layer of linear neurons.

As the transmitted signal is complex, a neural network algorithm has been generalized to neural network with complex weights [9]. The network's inputs are the columns of the matrix S, and the desired outputs are the same data as the inputs. The weights are adjusted according to a backpropagation algorithm [10], which minimizes the mean square error between the network outputs and the desired ones. Contrary to classical use of neural networks, the useful information is not the outputs of the network, but the weights. In fact we recover the spreading sequence in the second layer of weights. That is the reason why there is not previous train, but a training at each experiment. Hence we impose a condition to the two vectors corresponding to the second layer of weights. The constraint does not allow the vectors to have energy in the same time at the same place. In this way, adding the two vectors gives us the spreading sequence used by the transmitter.

3.3 Evaluation of the results

As the weights are complex, we recover the spreading code with a phase indetermination. It is not a problem, because in any transmission system symbols phase is always indeterminate on the receiver side. Anyway, in our application, it can be useful to normalize the phase for results interpretation. The phase is calculated according to the expected sequence, to visualize the results, as stated below :

let us note V the spreading code found with the neural network, we visualize $\hat{V}$ such as $\hat{V} = \mathrm{Re}\{Vz\}$, where $z = \dfrac{V^{*T}.H}{\|V\|^2}$, with H the true sequence.

4 Illustrative results

In many spread spectrum transmission systems, the spreading code is real when the channel effects are omitted, then we introduce several results with real sequences, treated with neural networks, the weights of which are first real and then complex. Then we study a transmission system, where the code and the network's weights are complex. To complete our work, we provide some results according to the signal to noise ratio (SNR) to the number of temporal windows N of duration T_s and according to the length of the spreading sequence.

4.1 Real sequence

4.1.1 With real weights

The studied PN sequence is a binary Gold code of length $P = 31$, and we consider a BPSK data modulation. The channel adds white, gaussian, centered, and real noise. The SNR is -5 dB (the signal power is less than the noise power).

Fig. 1 shows the code used by the transmitter.

in "Systems and Control: Theory and Applications", published by WSES, 2000, pp. 257-262, ISBN 960-8052-11-4

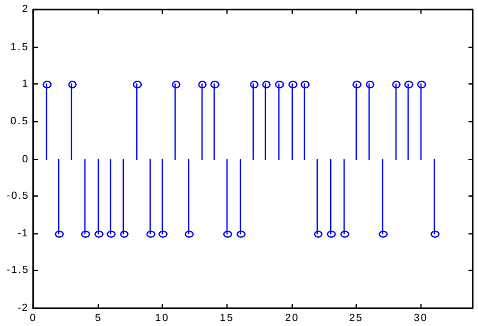

Fig. 1 : *Transmitter's PN code*

The first weight vector is shown on Fig. 2

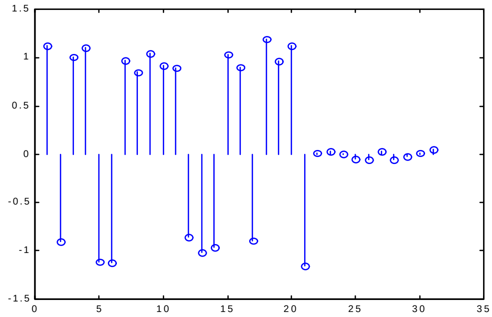

Fig. 2 : *First weight vector*

Fig. 3 shows the second weight vector.

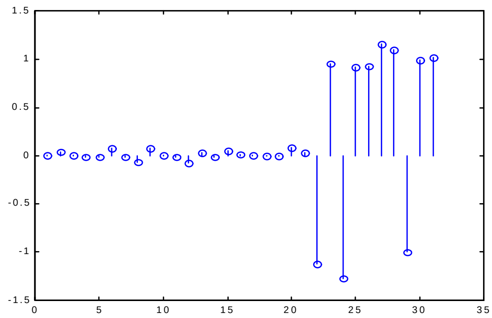

Fig. 3 : *second weight vector*

The first weight vector corresponds exactly to the end of the spreading sequence, whereas the end of the second weight vector corresponds to the opposite of the beginning of the code. Moreover we can observe that the constraint imposed to the vectors is well respected. There is only a problem of sign between the vectors, to recover the spreading code, we have to add the first one, with the opposite of the second one. It is a problem of phase indetermination.

4.1.2 **With complex weights**

The PN sequence is still a Gold code. It is the same as previously, and we consider now a QPSK modulation, damaged with a white, gaussian, centered and complex noise. The SNR is -10 dB.
To visualize the results, we used the technique of phase normalization.

Fig. 4 shows the estimated sequence (sum of the two weight vectors).

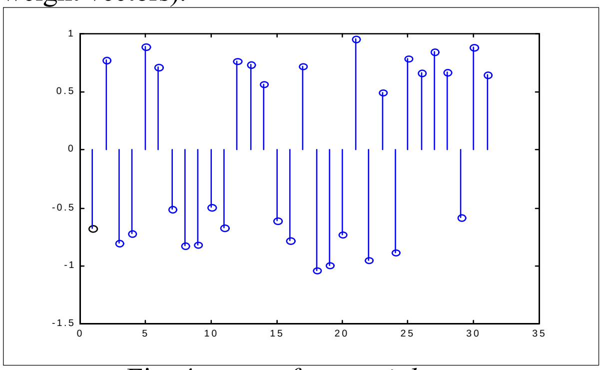

Fig. 4 : *sum of two weight vectors*

In comparison with Fig. 1, we recover exactly the spreading sequence, with a shift of ten positions left, because the received signal is not synchronized.

4.2 complex sequence

Let us now consider a complex sequence, the real and the complex parts of which are a Gold sequence. The information bearing signal is still a QPSK modulation, and the SNR is -10 dB. In this case we have to recover the real and imaginary parts of the sequence.

Fig. 5 represents the real part of the code.

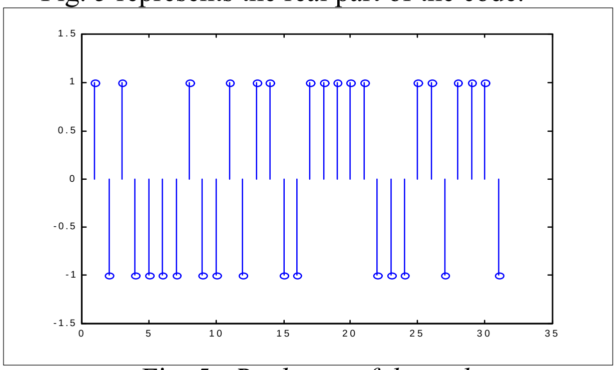

Fig. 5 : *Real part of the code*

and Fig. 6, the imaginary part

in "Systems and Control: Theory and Applications", published by WSES, 2000, pp. 257-262, ISBN 960-8052-11-4

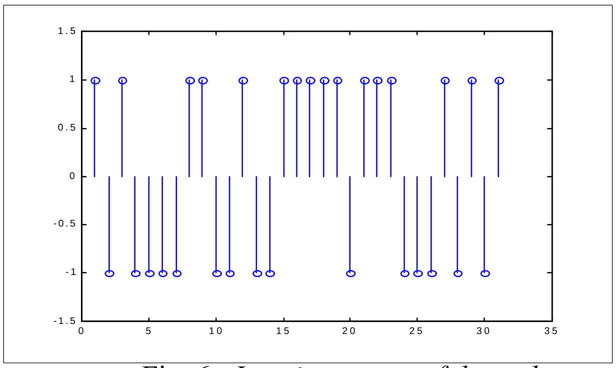

Fig. 6 : *Imaginary part of the code*

Fig. 7 and 8 represent the results of the neural network estimation (respectively real and imaginary parts of the weights).

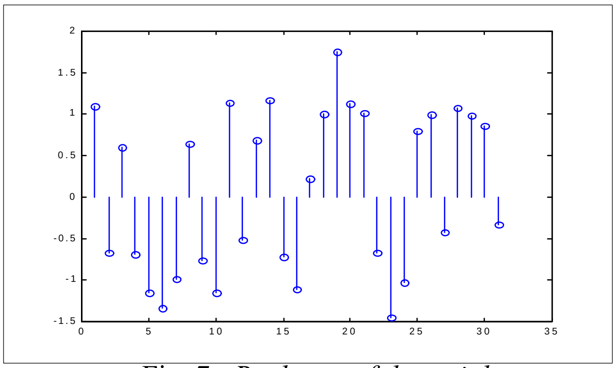

Fig. 7 : *Real part of the weights*

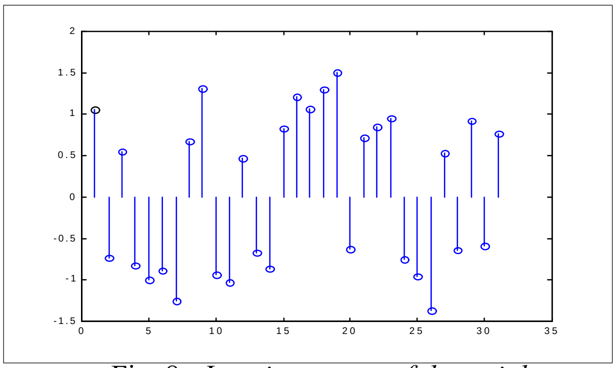

Fig. 8 : *Imaginary part of the weights*

If we compare Fig. 5 and Fig. 7, the signs of real part of the weights correspond exactly with the real part of the spreading sequence. Furthermore the signs of the imaginary part of the weights correspond exactly to the imaginary part of the code (Fig. 6 and Fig. 8).

4.3 Performances of the method

Here we introduce some tables summarizing the performances of our method.

4.3.1 Influence of the number of windows in the studied signal

We study the influence of the number of temporal windows N included in the signal used to estimate the spreading sequence.

For this experiment, the sequence length is $P = 31$, the modulation is a BPSK and the SNR is equal to -12 dB.

N	50	100	150	200
nb_errors	5	2	0	0

Table 1 :*Influence of the number of windows N*

nb_errors is the number of sign errors in the recovered sequence . When N increases, the results are improved.

4.3.2 Influence of the sequence length and the SNR

The modulation signal is a QPSK filtered at the transmitter and the receiver sides, the spreading sequence is a complex code, and we study the number of errors with respect to the SNR for several sequence lengths. For our experiment, we consider sequences of length $P = 31,\ 63,\ 127$, the real and imaginary parts of which are different Gold sequences, and we have N = 200. We assume, for simplicity, that $T_s = PT_c$. So the signal to noise ratio on the correlator output can be expressed as a function of the signal to noise ratio on the correlator input :

$$SNR_{out} = P\ SNR_{in}$$

If we express the signal to noise ratio in dB :

$$SNR_{out} = 10\log_{10}(P) + SNR_{in}$$

Hence the error probability per symbol Pe can be written as :

$$Pe = \frac{1}{2} erfc\left(\sqrt{\frac{SNR_{out}}{2}}\right)\left[2 - \frac{1}{2} erfc\left(\sqrt{\frac{SNR_{out}}{2}}\right)\right]$$

with $erfc(x) = \int_x^{+\infty} \frac{2}{\sqrt{\pi}} e^{-t^2} dt$

This shows that the performance of a transmitted spread spectrum signal is better with long sequences than with short ones.

Here are our results for different sequence lengths and SNR_{in} (dB)

	P = 31			P = 63			P = 127		
SNR	-2	-3	-4	-5	-6	-7	-9	-10	-11
errors	0	1	4	0	1	2	0	0	2

Table 2 : *Influence of sequence length*

errors are the number of signs errors in the sequence estimated by the neural network.

We observe, that the results are improved, when the sequence is longer. We get a gain of 3 dB when we use a sequence length equals to 63 rather than 31, or a sequence length equals to 127 rather than 63, which corresponds about to : ($10\log_{10}(63) - 10\log_{10}(31)$) and ($10\log_{10}(127) - 10\log_{10}(63)$). The results follow approximately the same law as the error probability per symbol.

5 Conclusion

In the context of spectrum surveillance, a method for identification of a spread spectrum transmitter PN sequence has been proposed. Experimental results have been provided and show good estimation results. Further work will include removal of sign or phase indecision.

References:

[1] D. Thomas Magill, Francis D. Natali, Gwyn P. Edwards, "Spread-Spectrum Technology for Commercial Applications", *Proceedings of the IEEE*, Vol. 82, No. 4, April 1994, pp. 572- 584.

[2] Raymond. L. Picholtz, Donald L. Schilling, Laurence B. Milstein, "Theory of Spread-Spectrum Communications - A Tutorial", *IEEE Transactions on Communications,* Vol. COM- 30, No. 5, May 1982, pp. 855- 884.

[3] Charles E. Cook, Howard S. Marsh, "An introduction to spread spectrum", *IEEE communications magazine,* March 1983, pp. 8-16.

[4] John G. Proakis, *Digital communications,* Third Edition, Mac Graw Hill International Editions, 1995.

[5] Dilip V. Sarwate, Michael B. Pursley, "Crosscorrelation Properties of Pseudorandom and Related Sequences", *Proceedings of the IEEE,* Vol. 68, No. 5, May 1980, pp. 593- 619.

[6] Alex W. Lam, Faith M. Özlütürk, "Performance Bounds for DS/SSMA Communications with Complex Signature Sequences", *IEEE Transactions on communications,* Vol. 40, No. 10, October 1992, pp. 1607- 1614.

[7] Michail K. Tsatsanis, Georgios B. Giannakis, "Blind Estimation of Direct Sequence Spread Spectrum Signals in Multipath", *IEEE Transactions on signal processing,* Vol. 45, No. 5, May 1997, pp. 1241- 1252.

[8] G. Burel, "Réseaux de Neurones en Traitement d'Images : des Modèles Théoriques aux Apllications Industrielles", *Thèse de Doctorat de l'Université de Bretagne Occidentale*, Brest , 1991.

[9] N. Rondel, "Réseaux de Neurones pour le Traitement d'Antenne et pour la Commande Référencée Capteur", *Thèse de Doctorat de l'Université de Bretagne Occidentale*, Brest , 1996.

[10] D. E. Rumelhart, G. E. Hinton, R. J. Williams, "Learning internal representations by error backpropagation", *Parallel Distributed Processing*, D. E. Rumelhart and J. L. Mc Clelland, chapter 8, Bradford book, MIT Press, 1986.