Exploiting symbolic dynamics in chaos coded communications with *maximum a posteriori* algorithm

F.J. Escribano, L. López and M.A.F. Sanjuán

A known *maximum a posteriori* (MAP) algorithm is adapted to decode chaotic signals sent over a noisy channel and get a low complexity MAP decoder that could be easily implemented. It is shown that this algorithm is useful for all chaotic encoding frameworks where symbolic dynamics could be applied and that the final bit error rate is better than that obtained with other usually employed *maximum likelihood* (ML) algorithms of similar complexity.

Introduction: In the past, much attention has been devoted to the task of estimating chaotic sequences affected by noise. Initial efforts were devoted to ML algorithms applied to specific systems such as chaos shift keying [1] or piecewise linear maps (PWLM) [2]. When the chaotic system admits the application of symbolic dynamics [3], a suboptimal ML Viterbi decoding is possible [4, 5].

MAP algorithms have been developed to estimate the initial condition in the case of PWLM [6]. Nevertheless, they are not easily adapted for the purposes of chaotic communications and they are not extensible to other kinds of encodings. As symbolic dynamics allow us to understand the decoder as a first-order Markov process, it is possible to adapt the MAP algorithm known as BCJR (after the initials of the authors) [7]. Although the resulting scheme is normally more complex compared to ML, we show that it is also possible to use it in a lower complexity and practical sliding-window framework that outperforms an ML Viterbi decoding of similar complexity.

Encoding: Though we focus on a very simple chaotic system for brevity's sake, it will be evident that all the following could be readily applied to other chaotic systems. To encode the binary sequence $\{b_n\}$, where $b_n \in \{0, 1\}$ and $n = 1, \ldots, N$, we will use the known Bernoulli shift map setup [4]. The binary sequence $\{b_n\}$ is independent and equiprobably distributed. The Bernoulli shift map iterates as follows:

$$x_{n+1} = f(x_n) = \begin{cases} 2x_n & \text{if } x_n \le 1/2 \\ 2x_n - 1 & \text{if } x_n > 1/2 \end{cases} \quad (1)$$

If we define $x_0 = \sum_{n=1}^{N} b_n 2^{-n}$, then the binary sequence is encoded into the chaotic sequence generated by x_0 and the information can be retrieved following $b_n = \lfloor x_n + (1/2) \rfloor$, where $\lfloor x \rfloor$ is the *floor* function, giving the closest integer below x. The probability density function (pdf) of the data generated is $p(x) = 1$ in [0, 1]. In other systems, where it is not possible to get a closed expression for the pdf, it is always possible to use a staircase approximation. The pdf is needed as *a priori* information for the decoder.

But this kind of encoding is not practical. We can encode blocks of D bits, following the discretisation $x'_n = \sum_{m=n}^{n+D-1} b_m 2^{-m+n-1}$. This can be seen as controlling the chaotic sequence with small perturbations or as encoding through symbolic dynamics [5]. When D is about 15–20, the quantisation effects are negligible, the samples keep the desired properties of a chaotic broadband noise-like signal and the discretised pdf is reasonably well approximated by the continuous pdf. We will refer to this chaotic controlled sequence simply as $\{x_n\}$.

Decoding algorithm: The received signal is $y_n = x_n + n_n$, where n_n is an additive white Gaussian noise (AWGN) with zero mean and power σ^2.

The setup for the chaotic BCJR algorithm needs the same definitions as the ML Viterbi algorithm in [4]. We split the [0, 1] interval into a series of non-overlapping intervals I_i with limits i/P and $(i+1)/P$ for $i = 0, \ldots, P-1$ and centre in $c_i = (i/P) + (1/2P)$. $P \le 2^D$, the number of intervals, is taken as a even number, so that the threshold point 1/2 is the upper point of one interval and the lower one of another. If we substitute the original sequence by the sequence of intervals where the corresponding sample lies, we get a symbolic representation of the sequence which also conveys the binary information and which can be described as a first-order Markov process, with a corresponding transition matrix T. The term t_{ij} in this matrix means the transition probability between the interval i and the interval j, and it depends on the quantisation grid and the form of the encoding. In the case of the Bernoulli shift map, each interval maps exactly into two contiguous intervals with equal probability. Note that this decoding framework can be applied even if symbolic dynamics at the encoder side does not match the symbolic dynamics at the receiver side.

We consider a decoding block of $L < N$ received symbols $\{y_n, \ldots, y_{n+L-1}\}$. We say that the state s_k at time $k = 1, \ldots, L$ is $s_k = i$ when $x_{n+k-1} \in I_i$. To build the BCJR algorithm, we have to calculate the following probability functions γ, α, β and λ as stated in [8]:

$$\begin{aligned} \gamma_n(i,j) &= Pr\{s_n = j, y_n | s_{n-1} = i\} \\ &= \sum_x Pr\{s_n = j | s_{n-1} = i\} \\ &\quad Pr\{x_n = x | s_{n-1} = i, s_n = j\} Pr\{y_n | x\} \end{aligned} \quad (2)$$

We substitute the possible values of x_n (which are 2^D) by the values of the centre of the intervals c_j (which are P). When $2^D = P$, we have an instance of Ungerboeck's trellis coded modulation, and it is not necessary to obtain the pdf to characterise the transitions. To calculate (2) we see that in a transition from $s_{n-1} = i$ to $s_n = j$, the only possible quantised encoder output is c_j, and $Pr\{x_n = x | s_{n-1} = i, s_n = j\}$ is 1 when $x_n \in I_j$ and is 0 in the rest of cases. $Pr\{s_n = j | s_{n-1} = i\}$ is the transition probability t_{ij} and $Pr\{y_n | c_j\} = [1/(\sigma\sqrt{2\pi})] e^{-(y_n - c_j)^2/2\sigma^2}$ is the channel output probability. Note that for systems with rate $R = 1/p$ less than one, we have only to include in this channel metric the p possible symbols for each transition. The algorithm operates as follows over each block of L symbols.

Calculate the probability function:

$$\gamma_k^n(i,j) = t_{ij} \frac{1}{\sigma\sqrt{2\pi}} e^{-(y_{n+k} - c_j)^2/2\sigma^2} \quad k = 1, \ldots, L \quad i, j = 0, \ldots, P-1 \quad (3)$$

Forward calculate the probability function:

$$\begin{aligned} \alpha_k^n(j) &= Pr\{s_k = j, \{y_n, \ldots, y_{n+k-1}\}\} \\ &= \sum_i \alpha_{k-1}^n(i) \gamma_k^n(i,j) \quad k = 1, \ldots, L \quad i, j = 0, \ldots, P-1 \end{aligned} \quad (4)$$

Backward calculate the probability function:

$$\begin{aligned} \beta_k^n(j) &= Pr\{\{y_{n+k}, \ldots, y_{n+L-1}\} | s_k = j\} \\ &= \sum_i \beta_{k+1}^n(i) \gamma_{k+1}^n(j,i) \quad k = L-1, \ldots, 0 \quad i, j = 0, \ldots, P-1 \end{aligned} \quad (5)$$

Finally compute the *a posteriori* probabilities:

$$\begin{aligned} \lambda_k^n(i) &= Pr\{s_k = i, \{y_n, \ldots, y_{n+L-1}\}\} \\ &= \alpha_k^n(i) \beta_k^n(i) \quad k = 1, \ldots, L \quad i = 1, \ldots, P \end{aligned} \quad (6)$$

To decode the bit at time n, we take the state i_{max} that maximises the probability $\lambda_1^n(i)$, and decode according to $b_n = \lfloor C_{i_{max}} + (1/2) \rfloor$. Both $\alpha_0^n(j)$ and $\beta_L^n(j)$ are usually initialised with $1/P$, $j = 0, \ldots, P-1$. When using this scheme in a sliding-window fashion, in superposing blocks of L symbols, we take as initial values for $\alpha_0^n(j)$ the resulting *a posteriori* values λ from the preceding decoding block $\lambda_1^{n-1}(j)$, and for $\beta_L^n(j)$ the α values of the forward algorithm $\alpha_L^n(j)$, reserving $1/P$ only for $\alpha_0^1(j)$. This ensures that the evidence of the received sequence is propagated to all the decoding blocks. The sliding-window proceeds forward just by taking the following block $\{y_{n+1}, \ldots, y_{n+L}\}$. In this way it is not necessary to store and process $N \times P$ values for the entire sequence. The overhead in calculations is compensated by the saving in memory and by the flexibility of the scheme (i.e. we can decode continuously).

Simulation results: In Fig. 1 we can see the results in terms of BER. $E_b/N_0 = \sigma_x^2/2\sigma^2$ and $\sigma_x^2 = 1/12$ is the power of the signal (ideally for $D \to \infty$). These results have been obtained with the sliding-window ML Viterbi algorithm [4] and the sliding-window MAP BCJR algorithm. The encoding discretisations are $D = 5$ and $D = 20$ bits per symbol. Note that the mismatch between the encoding and decoding discretisation ($D = 20$ and $P = 16, 32$), does not seem to affect the BER, compared with the matched case ($D = 5$ and $P = 32$).

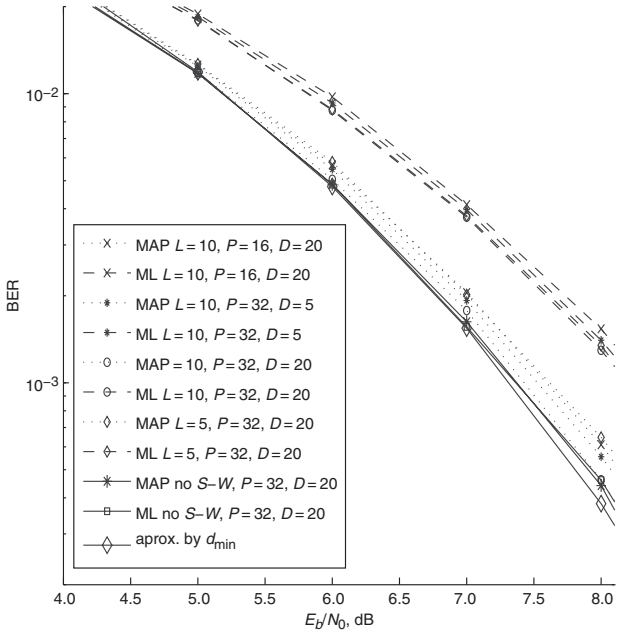

Fig. 1 *BER performance for different sets of parameters*

The sliding-window size is taken as $L=5$ and $L=10$. The results for the ML Viterbi and BCJR MAP decoding for the entire block (no sliding-window) are shown for comparison. In each case the MAP sliding-window algorithm yields better results than the ML sliding-window algorithm, while not differing much in complexity (which grows in both cases as $O(LP)$). The BCJR algorithm keeps closest to the limit calculated in [8] specially the case $L=10$, $P=32$, $D=20$, compared with cases with lower L or lower P, which hints that both increasing L or P lead to better performance and shows that the maximum attainable performance can be reached with limited complexity (keeping L fixed and increasing P or vice versa). The reason for the good behaviour of the BCJR algorithm as presented here is that it propagates efficiently the evidence throughout the sequence. Note also the relatively low effect of increasing L or P over ML results.

Conclusions: We have adapted the MAP BCJR decoding algorithm for a whole class of chaos-channel encoded signals under assumption of symbolic dynamics at the decoder side. We have shown that the BER performance attained by our MAP framework is generally better than the performance achieved with other popular and previously tested algorithms. The principles shown are readily extended to any other kind of encoding with symbolic dynamics.

It has to be noted that with the BCJR algorithm we get probabilistic soft estimates of the data, and this is most useful in some applications, such as the ones applying the iterative decoding philosophy of the so-called turbo codes. We expect that concatenating chaotic encoding systems we could make this kind of system comparable to the practical ones used in communications.

Acknowledgment: We acknowledge financial support from the Spanish Ministry of Science and Technology under project number BFM2003-0381.

16 March 2006
Electronics Letters online no: 20060649
doi: 10.1049/el:20060649

F.J. Escribano and M.A.F. Sanjuán (*Nonlinear Dynamics and Chaos Group, Departamento de Matemáticas y Física Aplicadas y Ciencias de la Naturaleza, Universidad Rey Juan Carlos, C/Tulipán s/n, 28933 Móstoles, Spain*)

L. López (*Laboratorio de Algoritmia Distribuida y Redes, Departamento de Informática, Estadística y Telemática, Universidad Rey Juan Carlos, C/Tulipán s/n, 28933 Móstoles, Spain*)

E-mail: francisco.escribano@urjc.es

References

1 Schimming, T., and Hasler, M.: 'Optimal detection of differential chaos shift keying', *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, 2000, **47**, (12), pp. 1712–1719
2 Pantaleon, C., Luengo, D., and Santamaria, I.: 'Optimal estimation of chaotic signals generated by piecewise-linear maps', *IEEE Signal Process. Lett.*, 2000, **7**, (8), pp. 235–237
3 Schweizer, J., and Schimming, T.: 'Symbolic dynamics for processing chaotic signals – II: Communication and coding', *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, 2001, **48**, (11), pp. 1283–1295
4 Kisel, A., Dedieu, H., and Schimming, T.: 'Maximum likelihood approaches for noncoherent communications with chaotic carriers', *IEEE Trans. Commun.*, 2001, **48**, (5), pp. 533–542
5 Ciftci, M., and Williams, D.: 'Optimal estimation and sequential channel equalization algorithms for chaotic communications systems', *EURASIP J. Appl. Signal Process.*, 2001, **4**, pp. 249–256
6 Pantaleon, C., *et al.*: 'Bayesian estimation of chaotic signals generated by piecewise-linear maps', *IEEE Trans. Signal Process.*, 2003, **83**, (3), pp. 659–664
7 Bahl, L.R., *et al.*: 'Optimal decoding of linear codes for minimizing symbol error rate', *IEEE Trans. Inf. Theory*, 1974, **20**, (2), pp. 284–287
8 Kozic, S., Oshima, K., and Schimming, T.: 'Minimum distance properties of coded modulations based on iterated chaotic maps'. Proc. NDES, Scuol, Switzerland, May 2003, pp. 141–144