

prepare a random lattice point v and precompute Bv in our idle time. When a message is given to be encrypted, only the encoding and addition of Bv and e are required, where Bv is already computed. The encoding of a message involves a simple translation of a binary number into $\pm\sigma$ and a random permutation. The latter can be processed in $O(n)$, although the above description allows for the inclusion of a multiplication of an $n \times n$ matrix and vector. Thus, the entire running time is $O(n)$.

In the original contribution, the authors suggest two encoding schemes. One has low bandwidth utilisation: it can send only $\log n$ bits of messages at a time. Our scheme is superior to that, and can send messages at a rate of $n/2$ bits.

Generalisation hint: The error vector may be quantised to a more delicate level. This multilevel quantisation permits a high information rate. That is, a single ciphertext contains much more information than in the case of two level quantisation. For an n dimensional lattice vector, $n/2$ bit data are encoded with two-level quantisation. With q -level quantisation, $n/2 \lg q$ bits of data will be encoded. However, multilevel quantisation requires multi-precision handling and it may cause decoding errors. Thus, special care must be taken when a multi-precision quantisation technique is used for analogue data encryption.

Conclusion: In this Letter, we have presented a method for encoding a message with a lattice-based public-key cryptosystem. The idea behind it is to embed a message in an error vector rather than in a lattice vector. The encoding scheme use quite a fast encryption procedure and has plaintext awareness. Only $O(n)$ operations are required to encrypt a message, which is a significant enhancement compared to the $O(n^2)$ of the original GGH scheme, and the $O(n^2)$ of RSA. Also, we have briefly described a multilevel quantisation technique for improving the information rate.

© IEE 1998

7 October 1998

Electronics Letters Online No: 19981589

DaeHun Nyang and JooSeok Song (Department of Computer Science, Yonsei University, SeodaemunGu, ShinchonDong 134, Seoul 120-749, Korea)

References

- GOLDREICH, O., GOLDWASSER, S., and HALVEI, S.: 'Public-key cryptosystems from lattice reduction problems'. Proc. CRYPTO'97, Santa Barbara, CA, 1997, pp. 112-131
- AJTAI, M.: 'Generating hard instances of lattice problems'. Proc. 28th STOC, Philadelphia, 1996, pp. 99-108
- AJTAI, M., and DWORK, C.: 'A public-key cryptosystem with worst-case/average-case equivalence'. Proc. 29th STOC, Texas, 1997, pp. 284-293
- VAN EMDE BOAS, P.: 'Another NP-complete problem and the complexity of computing short vectors in a lattice'. Report 81-04, Mathematische Instituut, University of Amsterdam, 1981
- GOLDWASSER, S., and MICALI, S.: 'Probabilistic encryption', *J. Comput. Syst. Sci.*, 1984, **28**, pp. 270-299

Modified turbo codes with low decoding complexity

Li Ping

A family of modified turbo codes with SPC precoding is presented. Compared with standard turbo codes, the new scheme results in considerably lower decoding cost yet can achieve nearly the same performance.

Introduction: In this Letter, a modified turbo code scheme is presented. The basic principle is to use SPC (single parity check) precoding to replace the puncturing technique [1] for rate adjustment. This greatly reduces the length of the convolutional code involved. It is also proposed that very simple constituent convolutional codes be used, e.g. those with four states, and that the number of constituent codes (so-called dimensions) be increased [2, 3] to maintain performance. Compared with the standard turbo codes

[1], the new scheme has considerably lower decoding cost yet can achieve nearly the same performance. The method is convenient for constructing median to high rate codes.

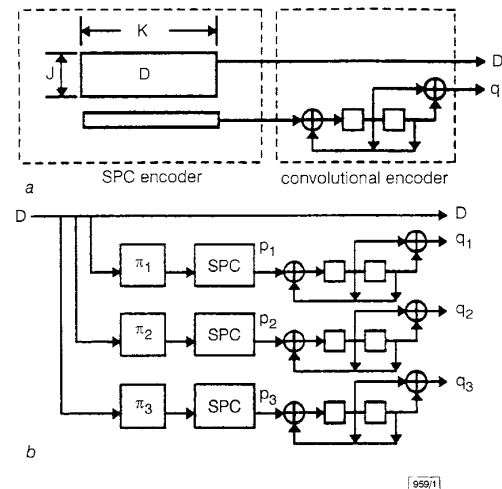


Fig. 1 SPC-convolutional encoder and three-dimensional SPC-turbo code

a SPC-convolutional encoder

b Three-dimensional SPC-turbo code

$\{\pi_i\}$ are interleavers and SPC stands for SPC encoder in Fig. 1a

Encoding and decoding principles: Fig. 1a illustrates the encoding principle of the constituent code. The information bits are arranged as a $J \times K$ array $\mathbf{D} = (D_{j,k})$. For convenience we will assume that \mathbf{D} is represented in the binary phase shift keying (BPSK) format over $\{+1, -1\}$. A parity check p_k is generated for every column of \mathbf{D} so that $\mathbf{D}_k \cup p_k$ contain an even number of -1 . The resultant vector $\mathbf{p} = \{p_k\}$ is used to drive a convolutional encoder with output sequence \mathbf{q} .

A three-dimensional concatenated code is illustrated in Fig. 1b based on that in Fig. 1a. It will be referred to as an SPC-turbo code. The information array \mathbf{D} is interleaved and encoded three times, producing \mathbf{q}_1 , \mathbf{q}_2 and \mathbf{q}_3 . The overall codeword is formed by $(\mathbf{D}, \mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3)$, with rate $R = J/(J+3)$. The principle can be generalised to N dimensions.

Decoding method: The global iterative decoder structure in [3] is adopted here for general multidimensional concatenated codes, which requires the MAP (maximum a posteriori) decoder for the constituent code. This can be accomplished for the code in Fig. 1a by treating it as a rate $J/(J+1)$ convolutional code, but the related trellis description has a high branch complexity (with 2^J branches emanating from every node). A more efficient method is now outlined. Only one dimension will be considered here so the dimension subscript for \mathbf{q} will be omitted.

Let $obs(\mathbf{c})$ be the noisy observation of a transmitted codeword \mathbf{c} in the BPSK format. Define the a priori LR (likelihood ratio) for a single bit c_n conditioned on $obs(c_n)$ as

$$\tilde{L}(c_n) \triangleq \frac{\Pr\{c_n = +1 | obs(c_n)\}}{\Pr\{c_n = -1 | obs(c_n)\}} \quad (1)$$

This is equivalent to the definition of LLR (logarithm of likelihood ratio) [1, 4]. Assuming that the a priori LR values are available for all the bits, the aim of the MAP decoding is to find the a posteriori LR for c_n as

$$L(c_n | obs(\mathbf{c})) \triangleq \frac{\Pr\{c_n = +1 | obs(\mathbf{c})\}}{\Pr\{c_n = -1 | obs(\mathbf{c})\}} \quad (2)$$

Eqn. 2 is evaluated for the code in Fig. 1a using the three step technique below. It is an exact solution but the proof will not be included due to space limitations.

Step (i): Let $\mathbf{D}_k = \{D_{j,k}\}$ be the k -column of \mathbf{D} . Compute $L(p_k | obs(\mathbf{D}_k))$ for $p_k, k = 1, 2, \dots, K$.

Step (ii): Use $L(p_k | obs(\mathbf{D}_k)), k = 1, 2, \dots, K$, as the a priori LR for \mathbf{p} . Apply MAP decoding to the rate $1/2$ convolutional code $\mathbf{p} \rightarrow (\mathbf{p}, \mathbf{q})$. This produces $L(p_k | obs(\mathbf{D}, \mathbf{q}))$. Define the extrinsic LR as $W(p_k) = L(p_k | obs(\mathbf{D}, \mathbf{q})) / L(p_k | obs(\mathbf{D}_k))$.

Step (iii): Perform MAP decoding for \mathbf{D}_k using $W(p_k)$ as the a priori LR for p_k .

In the above, step (ii) can be implemented by the standard technique [1]. Simple rules for implementing steps (i) and (iii) will now be given. Recall that $\mathbf{D}_k \cup p_k$ contains an even number of -1s and p_k is not transmitted (i.e. $L(p_k) = 1$). Thus

$$L(p_k | \text{obs}(\mathbf{D}_k)) = \text{plr}(\mathbf{D}_k) \quad (3)$$

where the plr (parity likelihood ratio) function is defined by

$$\text{plr}(\mathbf{D}_k) \triangleq \frac{\Pr\{\mathbf{D}_k \text{ even} | \text{obs}(\mathbf{D}_k)\}}{\Pr\{\mathbf{D}_k \text{ odd} | \text{obs}(\mathbf{D}_k)\}} \quad (4)$$

In eqn. 4, ' \mathbf{D}_k even' (' \mathbf{D}_k odd') indicates that \mathbf{D}_k contains an even (odd) number of -1s. If \mathbf{D}_k contains only one bit $D_{1,k}$, then $\text{plr}(D_{1,k}) = L(D_{1,k})$. Otherwise \mathbf{D}_k can be partitioned into two non-overlapping subsets: \mathbf{a} and \mathbf{b} . It can be shown that

$$\text{plr}(\mathbf{D}_k) = f(\mathbf{a}, \mathbf{b}) \triangleq \frac{\text{plr}(\mathbf{a})\text{plr}(\mathbf{b}) + 1}{\text{plr}(\mathbf{a}) + \text{plr}(\mathbf{b})} \quad (5)$$

Thus, step (i) can be realised by recursively generating $A_{j,k} = f(A_{j-1,k}, L(D_{j,k}))$, $j = 2, 3, \dots, J$, with initial values $A_{1,k} = \text{plr}(D_{1,k}) = L(D_{1,k})$. Then $L(p_k | \text{obs}(\mathbf{D}_k)) = \text{plr}(\mathbf{D}_k) = A_{J,k}$. Similarly, step (iii) can be solved as follows. Let $B_{j+1,k} = W(p_k)$ and $A_{-1,k} = 1$. Compute $B_{j,k} = f(B_{j+1,k}, L(D_{j,k}))$, $j = J, J-1, \dots, 2$. Then $L(D_{j,k} | \text{obs}(\mathbf{D}, \mathbf{q})) = L(D_{j,k})f(A_{j-1,k}, B_{j+1,k})$, $j = J, J-1, \dots, 1$.

The total cost of steps (i) and (iii) is about seven multiplications and three additions per information bit (ignoring add-by-1). For small values of J , it is slightly less.

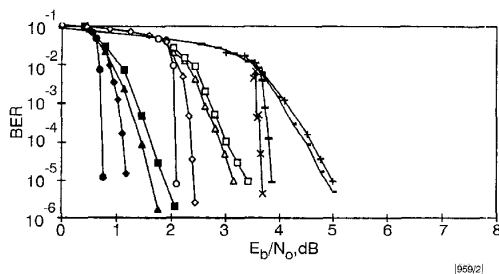


Fig. 2 Performances for three-dimensional, four-state turbo-SPC codes

R : coding rate; F : interleaver length (i.e. $F = J \times K$); i : iteration number

Limits ($R, E_b/N_0$): (0.5, 0.2dB), (0.75, 1.6dB)

Parameters for SPC encoding (R, J): (0.5, 3), (0.75, 9) and (0.9, 27)

- $R = 0.5, F = 59319, i = 18$
- ◆ $R = 0.5, F = 59319, i = 6$
- ▲ $R = 0.5, F = 1024, i = 18$
- $R = 0.5, F = 1024, i = 6$
- $R = 0.75, F = 59319, i = 18$
- △ $R = 0.75, F = 59319, i = 6$
- ◇ $R = 0.75, F = 1024, i = 18$
- $R = 0.75, F = 1024, i = 6$
- × $R = 0.9, F = 59319, i = 18$
- $R = 0.9, F = 59319, i = 6$
- $R = 0.9, F = 1024, i = 18$
- + $R = 0.9, F = 1024, i = 6$

Simulation results: Fig. 2 contains the simulation results for three-dimensional, four-state turbo-SPC codes in additive white Gaussian noise channels. This configuration is selected as a compromise between performance and complexity. The constituent convolution encoder is characterised by $(1+x)/(1+x+x^2)$. The interleaving schemes are selected empirically. For frame length $F = 59319$, an optimised random interleaving technique is used. For $F = 1024$, the three interleavers are given by $\pi_1 = \text{unity}$ (no interleaving), $\pi_2 = 32 \times 32$ block interleaver and $\pi_3 = \text{pseudo-random interleaver}$ of [1]. Overall, the performances shown are nearly the same as those for the 16-state standard turbo codes using the puncturing technique [1, 5]. For all the three rates shown, the length 59319 codes can achieve $\text{BER} = 10^{-5}$ at ~ 0.5 dB away from the theoretical limits.

Discussions and conclusions: The MAP decoding cost, in terms of operations involved, for a convolutional code is proportional to the product of state number and code length. For the SPC-turbo codes considered above, the state number is only 4. Also notice

that the total length L (for all the dimensions) of the convolutional code in Fig. 1b is $L = 3F/J$. Since $R = J/(J+3)$, we have $L = (R^{-1}-1)F$, which reduces when R increases toward 1 (e.g. $L = F$ for $R = 0.5$ and $R = F/9$ for $R = 0.9$). As a comparison, a standard turbo code using puncturing technique [1, 5] involves the convolutional code length of $2F$, regardless of R . Including steps (i) and (iii), the decoding cost of an SPC-turbo code is ~ 4 (for rate ≈ 0.5) to 8 (for rate ≈ 1) times less than a 16-state standard turbo code. A recently proposed method using true 2/3 constituent code [6] can also lead to a cost reduction by half compared to standard turbo codes.

© IEE 1998

28 September 1998

Electronics Letters Online No: 19981537

Li Ping (Department of Electrical Engineering, City University of Hong Kong, 83 Tat Chee Avenue, Kowloon, Hong Kong)

E-mail: celiping@cityu.edu.hk

References

- BERROU, C., GLAVIEUX, A., and THITIMAISHIMA, P.: 'Near optimum error-correcting coding and decoding: Turbo-codes', *IEEE Trans. Commun.*, 1996, **COM-44**, pp. 1261-1271
- DIVSALAR, D., and POLLARA, F.: 'Multiple turbo codes', Proc IEEE Milcom-95, 1995, pp. 279-285
- LI PING, CHAN, S., and YEUNG, K.L.: 'Iterative decoding of multidimensional concatenated single parity check codes', Proc. ICC-98, 1998, pp. 136-140
- HAGENAUER, J., OFFER, E., and PAPKE, L.: 'Iterative decoding of binary block and convolutional codes', *IEEE Trans.*, 1996, **IT-42**, pp. 429-445
- ACIKEL, O.F., and RYAN, W.E.: 'High rate turbo codes for BPSK/QPSK channels', Proc. ICC-98, 1998, pp. 422-427
- BERROU, C.: 'Some clinical aspect of turbo codes', Proc. Int. Symp. Turbo Codes, Brest, France, 1997, pp. 26-31

Classification of odour samples from multisensor array using new linguistic fuzzy method

B. Lazzarini, A. Maggiore and F. Marcelloni

A new method for the fuzzy classification of odour samples that are obtained from an array of conducting polymer sensors is proposed. Linguistic expressions describing the response of both individual sensors and the sensor array to each chemical are derived from a fuzzy model of the sensor data. Experimental results confirming the proposed method are also included.

Introduction: The most common method for classifying odours involves the use of a panel of human assessors [1]. However, owing to such factors as state of health, mood and habits, human assessment cannot be considered as objective and consistent. Automated odour discrimination would therefore be desirable. Furthermore, it can be extremely useful in situations where toxic or obnoxious samples have to be evaluated. In this Letter we present a new method, called BAF, for the automated classification of odour samples. BAF classifies signals produced by an array of conducting polymer sensors with partially overlapping sensitivities. Fig. 1 shows the responses of one sensor to a given chemical in 11 repeated experiments. It can be seen that the signals are noisy and that there is a strong drift in their amplitudes. Nevertheless, the general shape of the signals does not change. BAF explicitly models the uncertainty present in sensor responses building a linguistic fuzzy model [2] for each pair sensor/chemical. The model describes in linguistic terms the shape of the sensor response to the chemical. Also, an independent fuzzy model is built to represent the dynamic range of the signals. When an unknown chemical has to be recognised, it is identified as being the chemical with the highest match with respect to shape and dynamic range.

Fuzzy partition of input and output spaces: For each response of a sensor i to a chemical j , the input space and the output space (i.e.