

References

- 1 BARKER, R.H.: 'Group synchronizing of binary digital systems' in JACKSON, W. (Ed.): 'Communication theory' (Butterworths, London, 1953), pp.273-287
- 2 GOLOMB, S.W., and SCHOLTZ, R.A.: 'Generalized Barker sequences', *IEEE Trans. Inf. Theory*, 1965, **IT-11**, (4), pp. 533-537
- 3 ZHANG, N., and GOLOMB, S.W.: 'Sixty-phase generalized Barker sequences', *IEEE Trans. Inf. Theory*, 1989, **IT-35**, (4), pp. 911-912
- 4 BÖMER, L., and ANTWEILER, M.: 'Polyphase Barker sequences', *Electron. Lett.*, 1989, **25**, (23), pp. 1577-1579
- 5 FRIESE, M., and ZOTTMANN, H.: 'Polyphase Barker sequences up to length 31', *Electron. Lett.*, 1994, **30**, (23), pp. 1930-1931
- 6 CHANG, N., and GOLOMB, S.W.: '7200-phase generalized Barker sequences', *IEEE Trans. Inf. Theory*, 1996, **IT-42**, (4), pp. 1236-1238
- 7 FRIESE, M.: 'Polyphase Barker sequences up to length 36', *IEEE Trans. Inf. Theory*, 1996, **IT-42**, (4), pp. 1248-1250
- 8 SOMAINI, U., and ACKROYD, M.H.: 'Uniform complex codes with low autocorrelation sidelobes', *IEEE Trans. Inf. Theory*, 1974, **IT-20**, pp. 689-691

Simplified MAP algorithm suitable for implementation of turbo decoders

W.J. Gross and P.G. Gulak

A simplification of the MAP decoder for use in turbo decoders is presented. The simplification eliminates the need for a ROM or multiplexor-tree lookup table and replaces it with a constant value. Simulations show that the performance of turbo decoders is not adversely affected by this simplification.

Introduction: Maximum *a posteriori* (MAP) decoding of convolutional codes has seen a resurgence of interest since the discovery of turbo codes in 1993 [1]. MAP decoders make optimum symbol-by-symbol decisions, as well as providing 'soft' reliability information which is necessary in concatenated decoding systems such as turbo decoders. The BCJR algorithm [2] is the most commonly used MAP algorithm in turbo decoding. The BCJR algorithm suffers several shortcomings which make it unsuitable for VLSI implementation, namely the requirement for multiplications and exponentiations. The concept of applying the Jacobi logarithm to simplify MAP decoders was first introduced by Erfanian and Pasupathy [3]. In the logarithmic form of these algorithms, exponentials disappear, multiplications become additions and additions become the MAX* operation (using Viterbi's notation [4]). In this Letter, we introduce a simplified version of the MAX* operation and demonstrate the performance of a turbo decoder using our simplification.

Log-BCJR algorithm: We will briefly describe the BCJR algorithm in the logarithmic domain, the log-BCJR algorithm. Refer to [2, 4, 5] for detailed derivations of the algorithm. Consider an information block **u** of *N* bits which take on values of +1 or -1. We encode the block to obtain the coded block of symbols **c**. After transmission through a channel we receive the block **y**. We describe a transition in the binary trellis of the code by its starting state *s'* at time *k-1* and its ending state *s* at time *k*. The output of the algorithm is the log-likelihood ratio of *a posteriori* probabilities:

$$L_k = \log \frac{P(u_k = +1|y)}{P(u_k = -1|y)} = \left(\text{MAX}_{\substack{(s',s) \\ u_k=+1}}^* (A_{k-1}(s') + G_k(s',s) + B_k(s)) \right)$$

$$- \text{MAX}_{\substack{(s',s) \\ u_k=-1}}^* (A_{k-1}(s') + G_k(s',s) + B_k(s)) \quad (1)$$

Following Erfanian and Pasupathy [3] we use the Jacobi logarithm:

$$\log(e^X + e^Y) = \text{MAX}(X, Y) + \log(1 + e^{-|X-Y|}) = \text{MAX}^*(X, Y) \quad (2)$$

Viterbi calls this the MAX* operation [4] denoting that it is essentially a MAX operation adjusted by a correction factor. For an AWGN channel, the branch metrics $G_k(s', s)$ are

$$G_k(s', s) = -\frac{1}{2\sigma^2} \|y_k - c_k(s', s)\|^2 \quad (3)$$

where $c_k(s', s)$ is the symbol associated with the transition (*s'*, *s*) in the convolutional encoder. Given that the trellis starts and ends in the all-zeros state, the forward state metrics are calculated by a forward recursion as

$$A_k(s) = \text{MAX}_{s'}^* (G_k(s', s) + A_{k-1}(s')) \quad k = 1, \dots, N-1 \quad (4)$$

with initial conditions:

$$A_0(0) = 0 \\ A_0(s) = -\infty \quad s \neq 0 \quad (5)$$

The reverse state metrics are calculated by a backwards recursion as

$$B_{k-1}(s') = \text{MAX}_s^* (G_k(s', s) + B_k(s)) \quad k = N, \dots, 2 \quad (6)$$

with initial conditions:

$$B_N(0) = 0 \\ B_N(s) = -\infty \quad s \neq 0 \quad (7)$$

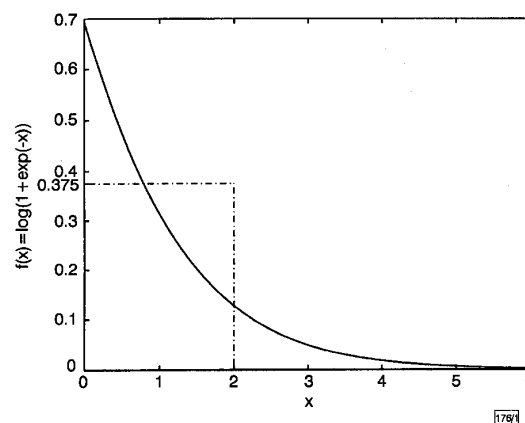


Fig. 1 Correction function and its approximation

— $f(x)$
- - - approx. $f(x)$

Simplified MAX* operation: In this Section, we discuss a simplified implementation of the MAX* operation for use in the log-BCJR algorithm. It is well-known that the correction factor in eqn. 2 can be approximated in a small lookup table with negligible effects on performance [3, 5]. We plot the correction function $f(x) = \log(1 + \exp(x))$ in Fig. 1. It has been reported that excellent results are obtained by restricting *x* to between 0.0 and 5.0 and storing eight values in the table [5]. We have implemented the log-BCJR algorithm using 8 bit two's complement integer arithmetic. Our numerical representation used five integer bits and three fractional bits and, therefore, the smallest value of $f(x)$ we can represent is 1/8, corresponding to a maximum value of *x* of ~2.0. Our simulations of a turbo decoder using the log-BCJR algorithm have shown that the function in Fig. 1 can be approximated by the following rule:

$$f(x) = \begin{cases} \frac{3}{8} & -2 \leq x < 2 \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

The implication of the above rule is that the lookup table for $f(x)$ can be reduced to a simple logic circuit which either adds or does not add a constant to the output of the maximum selection circuit. The simplified circuit is shown in Fig. 2. An eight-location lookup table can be implemented in CMOS using 60 transistors, assuming that both $X-Y$ and $Y-X$ are available from the subtractor circuit.

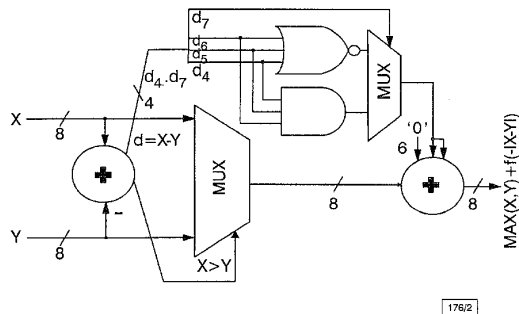


Fig. 2 Simplified MAX* operation

The simplified circuit can be implemented using 20 transistors and only requires the difference $X-Y$. Synthesis using standard cells in $0.5\mu\text{m}$ CMOS results in an area saving of 40%. The savings are even more pronounced when considering FPGA or DSP implementations. In these cases, the multiplexor tree required for the eight-location lookup table cannot be realised as efficiently. The lookup table may require storage in RAM or ROM which can create a memory bottleneck. In addition, the subtractor circuits used are not likely to provide both differences required and therefore two subtractors or an absolute value circuit is required. The simplified rule does not have any of these restrictions and is well suited to FPGA or DSP implementations.

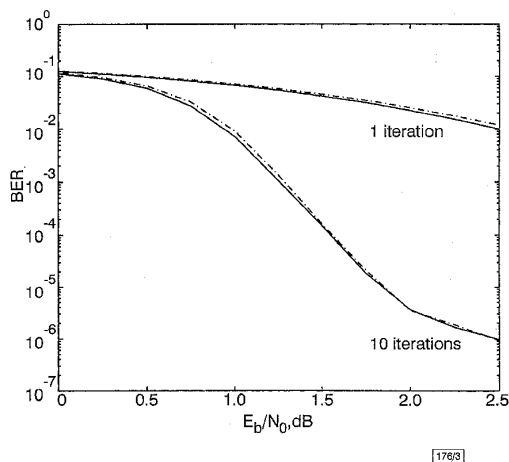


Fig. 3 Rate 1/2, 4-state, blocksize 1024 turbo code performance
— floating point
--- quantised, new MAX*

Performance: Fig. 3 shows the performance of a four-state rate 1/2 turbo code (polynomials 7/5) with a block length of 1024 for one and ten iterations using our simplified MAX* operation and 8 bit metrics compared to a floating point simulation. There is a negligible performance loss of 0.03dB at high bit error rates, but at the interesting bit error rate region around 10^{-5} and 10^{-6} there is practically no difference in performance.

Conclusions: We have introduced a simplification of the MAX* operation for the log-BCJR algorithm which replaces the lookup table with a constant value. Simulations show that turbo code performance is not adversely affected.

Acknowledgments: The authors wish to thank V. Gaudet for helpful comments. Funding from the Natural Sciences and Engineering Council of Canada is gratefully acknowledged.

© IEE 1998

Electronics Letters online No: 19981120

22 June 1998

W.J. Gross and P.G. Gulak (Department of Electrical and Computer Engineering, University of Toronto, 10 King's College Road, Toronto, Ontario, M5S 3G4 Canada)

References

- 1 BERROU, C., GLAVIEUX, A., and THITIMAJSHIMA, P.: 'Near Shannon limit error-correcting coding and decoding: Turbo codes'. Proc. IEEE Int. Conf. Communications, 1993, pp. 1064-1070
- 2 BAHL, L.R., COCKE, J., JELINEK, E., and RAVIV, J.: 'Optimal decoding of linear codes for minimising symbol error rate', *IEEE Trans. Inf. Theory*, 1974, pp. 284-287
- 3 ERFANIAN, J.A., and PASUPATHY, S.: 'Low-complexity parallel-structure symbol-by-symbol detection for ISI channels'. IEEE Pacific Rim Conf. Communications, Computers and Signal Processing, 1989, pp. 350-353
- 4 VITERBI, A.J.: 'An intuitive justification and a simplified implementation of the MAP decoder for convolutional codes', *IEEE J. Sel. Areas Commun.*, 1998, 16, (2), pp. 260-264
- 5 ROBERTSON, P., and HOEHER, P.: 'Optimal and sub-optimal maximum a posteriori algorithms suitable for turbo decoding', *Eur. Trans. Telecommun.*, 1997, 8, (2), pp. 119-125

Measurement of small resistance variations using $\Sigma\Delta$ technique

S. Gal, G. Cathébras and Y. Bertrand

A novel circuit is proposed for measuring small resistance variations. The method uses a $\Sigma\Delta$ modulator to balance a current controlled resistor bridge. This allows direct analogue to digital conversion without using a high precision instrumentation amplifier.

Introduction: To make a low-cost microelectromechanical system (MEMS) on silicon, it is possible to use a standard CMOS process for the electronic part, and very simple post-processing for freeing mechanical structures (bridges, cantilevers etc.) that are only made of the upper layers of the integrated circuit (silicon dioxide, polysilicon, metal) [1]. Since the fabrication of the MEMS starts with a standard CMOS process, the polysilicon is not doped to maximise its piezoresistivity, but to minimise its resistivity. Consequently, a sensor, for instance an accelerator sensor, will exhibit a very small relative variation of the resistances of its constraint gauges.

In this Letter, we present the design and results obtained with a first-order sigma delta modulator associated with a current-controlled bridge containing two variable resistors (or one variable and one fixed). The originality of the proposed structure lies in the fact that the value of the resistors is not obtained through measurement of the imbalance voltage of the bridge, but through the amount of current that must be added to the static branches of the bridge to make it balanced. Since this is accomplished through the feed-back loop of the sigma delta modulator, this leads to the compensation (due to the high DC gain of the integrator) of many sources of error, and avoids the use of a high precision component, or instrumentation amplifier.

Resistance variation measurement: The heart of the proposed structure, (Fig. 1) is the current controlled resistor bridge. The differential voltage is given by $V_d = V_{dd} - (R_1 + R_2) I$. We set $R_0 = R_1 + R_2$ and $V_{dd} = R_0 I_0$ for the balanced conditions where $V_d = 0$. If we introduce a variation ΔR on $R_1 + R_2$, defined by $\Delta R = (R_1 + R_2) - R_0$, we have to set I to $I_0 + \Delta I$ such that V_d is 0 again. We obtain

$$V_{dd} - R_0 I_0 = V_{dd} - (R_0 + \Delta R)(I_0 + \Delta I) \quad (1)$$

$$\frac{\Delta R}{R_0} = -\frac{1}{1 + \frac{I_0}{\Delta I}} \quad (2)$$