ON MISMATCH ERRORS IN ANALOG-VLSI ERROR CORRECTING DECODERS

Felix Lustenberger and Hans-Andrea Loeliger

Signal and Information Processing Laboratory, Swiss Federal Institute of Technology,
Sternwartstrasse 7, 8092 Zürich, Switzerland. E-mail: [lustenbe, loeliger]@isi.ee.ethz.ch

ABSTRACT

A new type of nonlinear analog transistor networks has recently been proposed for "turbo" decoding of error correcting codes. However, the influence of various nonidealities on the performance of such analog decoders is not yet well understood. The paper addresses the performance degradation due to transistor mismatch. Some analytical results are derived that allow to compare the accuracy of analog decoders with that of digital decoders. Moreover, these results enable to incorporate transistor mismatch into fast high-level simulations.

1. INTRODUCTION

Error correcting codes are a central part of most modern communication systems. The best known such codes — best in the sense of performing closest to the theoretical Shannon limit — are turbo codes [1], low-density parity check codes [2, 3], and variations thereof. Such codes are decoded by "probability propagation"—a version of the generic sum-product algorithm [4]—or variations thereof. Concise recent reviews of such codes and their decoding are [5] and [6], respectively.

In contrast to many other signal processing tasks in a communications receiver, the decoding of error correcting codes has always been implemented digitally. However, it was recently observed that the probability propagation algorithm can be naturally mapped onto elementary transistor circuits [7–9]. With these circuits, analog-VLSI decoders for turbo codes, low-density parity check codes, and similar codes can be built. The advantage of such analog decoders over digital decoders are higher operating speed (for some fixed power consumption) or lower power consumption (for some fixed operation speed) or both; it is expected that these gains can amount to two orders of magnitude [9–11].

Very similar analog decoders were investigated by Hagenauer *et al.* [12–14]. Alternative approaches to analog decoding include the analog Viterbi decoders of [15–18]; for further references see [9].

Clearly, the performance of such analog decoders will be affected by all sorts of nonidealities. Most such effects are not easily quantified. In principle, all such effects could be studied by SPICE-level Monte-Carlo simulations, but such simulations are far too time consuming to be of much use. Better methods are definitely needed.

The present paper focusses on the effects of transistor mismatch, which appears to be the primary influence limiting the precision of the computation in such decoders [11]. We will see that, for individual circuit modules, transistor mismatch affects the represented probabilities in an analytically tractable way. This result is used in a high-level simulation tool, which enables us to study the global effects of transistor mismatch by simulations that are only slightly slower than the numerical simulations of an ideal decoder.

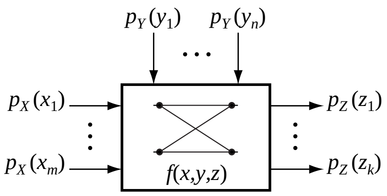

Figure 1: Generic sum-product module.

The paper is organized as follows. In Sections 2 and 3, we briefly review the elementary circuit module of the analog decoding networks. The effects of mismatch are analyzed, first in Section 4 for an individual transistor, then in Section 5 for a whole circuit module. Some simulation results are presented in Section 6, and some concluding remarks are offered in Section 7.

2. REVIEW OF BASIC SUM-PRODUCT CIRCUIT

Decoding by probability propagation can be decomposed into building blocks of the type shown in Fig. 1 [9]. Such a building block takes as input two probability distributions, p_X defined on some set $\mathcal{X} = \{x_1, \ldots, x_m\}$ and p_Y defined on some set $\mathcal{Y} = \{y_1, \ldots, y_n\}$, and computes as output the probability distribution p_Z defined on $\mathcal{Z} = \{z_1, \ldots, z_k\}$ by

$$p_Z(z) = \gamma \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p_X(x) p_Y(y) f(x, y, z), \tag{1}$$

where f is a $\{0,1\}$-valued function and where γ is an appropriate scale factor that does not depend on z. By varying m, n, k, and the function f, a large class of sum-product modules is obtained.

As shown in [7, 9], any such sum-product module can be realized by the transistor circuit shown in Fig. 2. The probability distributions p_X, p_Y, and p_Z are represented by current vectors, e.g., $[I_{x,1}, \ldots, I_{x,m}] \triangleq I_s[p(x_1), \ldots, p_X(x_m)]$, with an arbitrary sum current I_s. The transistors will be modeled as exponentially behaving voltage controlled current sources defined by

$$I_{\mathrm{C}} = I_0(T) e^{V_{\mathrm{BE}}/nU_{\mathrm{T}}}, \tag{2}$$

where T is the absolute temperature and U_{T} is the thermal voltage kT/q. The factor n is the "emission coefficient," an indicator of imperfect emission of electrons, which is usually close to one.

0-7803-6685-9/01/$10.00©2001 IEEE

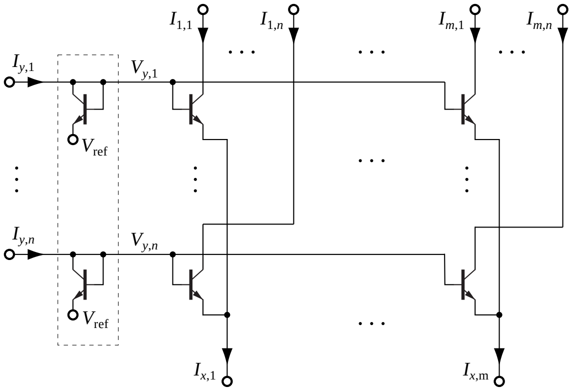

Figure 2: Products circuit.

Given this transistor model, the circuit of Fig. 2 computes the output currents $I_{i,j}$ according to

$$I_{i,j} = I_z (I_{x,i}/I_x)(I_{y,j}/I_y) \tag{3}$$

with $I_x \triangleq \sum_{i=1}^{m} I_{x,i}$, $I_y \triangleq \sum_{j=1}^{n} I_{y,j}$, and $I_z \triangleq \sum_{i=1}^{m}\sum_{j=1}^{n} I_{i,j} = I_x$. The circuit thus simultaneously computes all products $p(x_i)p(y_j)$ of (1). In order to complete the computation of (1), for each $z \in \mathcal{Z}$, those product terms $p_X(x)p_Y(y)$ for which $f(x,y,z)$ is nonzero have to be added. This is easily accomplished by connecting the corresponding wires, relying on Kirchhoff's current law. By adding current mirrors at the inputs and outputs, the modules become freely cascadable [7, 9].

3. INPUT VOLTAGES AND THE LOG-LIKELIHOOD REPRESENTATION

In Section 4, we will express the effects of transistor mismatch in terms of voltages. It should therefore be pointed out that, in the circuit of Fig. 2, voltages represent logarithms of probabilities [9], also called log-likelihoods. The diode-connected transistors within the dashed box in Fig. 2 convert the input current vector $(I_{y,1},\ldots,I_{y,n}) = (I_y\, p_Y(y_1),\ldots,I_y\, p_Y(y_n))$ into a voltage vector $(V_{y,1},\ldots,V_{y,n})$ with voltages

$$V_{y,j} = nU_{\mathrm{T}} \log(p_Y(y_j)) + nU_{\mathrm{T}} \log(I_y/I_0) + V_{\mathrm{ref}}. \tag{4}$$

By omitting the diode-connected transistors within the dashed box, the circuit could be converted into one with voltage inputs instead of current inputs. The voltage input vector $(V_{y,1},\ldots,V_{y,n})$ would then be given by

$$V_{y,j} = nU_{\mathrm{T}} \log(p_Y(y_j)) + V_{\mathrm{offs}}, \tag{5}$$

where V_{offs} can be chosen freely. This approach was chosen by Moerz *et al.* [13, 14] whose analog decoding networks are also based on the circuit shown in Fig. 2, but without the transistors in the dashed box. It should be noted, however, that the relation (5) between voltages and probabilities depends on the absolute temperature.

Digital decoders often work with logarithms of probabilities rather than with the actual probabilities. The quantization in such log-domain digital decoders corresponds to a quantization of the input voltage (5) in an analog decoder. Conversely, a limited accuracy in the voltage (5) can be translated into an equivalent resolution (in bits) of a log-domain digital decoder.

4. SINGLE-TRANSISTOR MISMATCH

The main effect of transistor mismatch is that the current I_0 in (2) is changed to $I_0(1+\varepsilon)$. The output current I_C is thus changed to

$$\begin{aligned} I_C(1+\varepsilon) &= I_0\, e^{\frac{V_{\mathrm{BE}}}{nU_{\mathrm{T}}}}\,(1+\varepsilon) \\ &= I_0\, e^{\frac{V_{\mathrm{BE}}+nU_{\mathrm{T}}\ln(1+\varepsilon)}{nU_{\mathrm{T}}}}. \end{aligned} \tag{6}$$

The relative current error is thus equivalent to an absolute error in V_{BE} of

$$\begin{aligned} \delta V_{\mathrm{BE}} &= nU_{\mathrm{T}} \ln(1+\varepsilon) && (7)\\ &\le nU_{\mathrm{T}}\varepsilon. && (8) \end{aligned}$$

For any given base-emitter voltage swing ΔV_{BE}, the quantity $\log_2(\Delta V_{\mathrm{BE}}/\delta V_{\mathrm{BE}})$ will be referred to as the equivalent digital resolution. As was explained in Section 3, it may may be interpreted as the resolution (in bits) of a log-domain digital decoder. Such digital decoders often operate with a resolution of 4 to 7 bits.

Some numerical values (for T=300K and n=1) are given in Table 1. Note that a voltage swing of $\Delta V_{\mathrm{BE}} = 300$ mV corresponds to a current with a dynamic range of 5 decades (60 mV per decade).

ε	δV_{BE}	resolution at	
		DR = 4 dec	DR = 5 dec
		$\Delta V_{\mathrm{BE}} = 240$ mV	$\Delta V_{\mathrm{BE}} = 300$ mV
1 %	0.258 mV	9.9 bits	10.2 bits
5 %	1.26 mV	7.6 bits	7.9 bits
10 %	2.47 mV	6.6 bits	6.9 bits
25 %	5.78 mV	5.4 bits	5.7 bits
50 %	10.50 mV	4.5 bits	4.8 bits

Table 1: Single-transistor mismatch and equivalent digital resolution.

5. MISMATCH IN THE BASIC TRANSISTOR MATRIX

We assume in the following that each transistor in the circuit of Fig. 2 is affected by transistor mismatch that changes I_0 into $I_0(1+\varepsilon)$. The corresponding error terms ε will be denoted ε_j, $j = 1,\ldots,n$, for the diode-connected transistors within the dashed box and $\varepsilon_{i,j}$, $i = 1,\ldots,m$, $j = 1,\ldots,n$, for the transistor in row j and column i of the transistor matrix. The key idea of the following analysis is that the mismatch errors can be expressed by a modified *input* probability distribution p_Y as illustrated by Fig. 3.

First, it is easy to see that mismatch of the diode-connected transistors within the dashed box is equivalent to a modified input distribution

$$\hat{p}(y_j) = \frac{(1+\varepsilon_j)\, p(y_j)}{\sum_{\ell=1}^{n}(1+\varepsilon_\ell)\, p(y_\ell)} \tag{9}$$

Second, by (6), the mismatch of the transistor in row j and column i is equivalent to an absolute error of $nU_{\mathrm{T}}\ln(1+\varepsilon_{i,j})$ in its base voltage $V_{y,j}$. Again by (6), this is in turn equivalent to a factor $1+\varepsilon_{i,j}$ in the input current $I_{y,j}$. For some fixed column i, the total mismatch error can thus be expressed by modifying the input

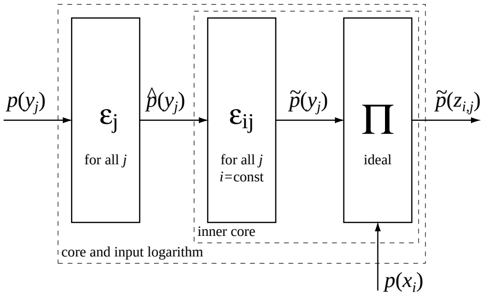

Figure 3: Backpropagation principle of the mismatch errors.

distribution p_Y to $\tilde{p}_Y$ defined by

$$\tilde{p}(y_j) = \frac{(1+\varepsilon_{i,j})\,\hat{p}(y_j)}{\sum_{k=1}^{n}(1+\varepsilon_{i,k})\,\hat{p}(y_k)} \qquad (10)$$

$$= \frac{(1+\varepsilon_{i,j})(1+\varepsilon_j)\,p(y_j)}{\sum_{k=1}^{n}(1+\varepsilon_{i,k})(1+\varepsilon_k)\,p(y_k)}. \qquad (11)$$

Combining this with (3), the actual normalized transistor matrix output current $I'_{i,j}/I_z$ is given by

$$I'_{i,j}/I_z = p_X(x_i)\tilde{p}_Y(y_j). \qquad (12)$$

This formulation is suitable for high-level simulation software. However, we will also be interested in the logarithmic error

$$\ln\frac{I'_{i,j}}{I_{i,j}} = \ln\frac{(1+\varepsilon_{i,j})(1+\varepsilon_j)}{\sum_{k=1}^{n}(1+\varepsilon_{i,k})(1+\varepsilon_k)\,p(y_k)} \qquad (13)$$

where $I_{i,j} = I_z\,p_X(x_i)p_Y(y_j)$ is the nominal output current. As in the previous section, this quantity can be used for comparisons with digital implementations.

6. SIMULATION RESULTS

In the simulations described in the following, the parameters ε of all transistors were considered as independent Gaussian random variables [19], truncated to the range $]-1,+1]$. The simulation results are divided into two parts. First, we consider the distribution of (13). Second, simulation results for a binary low-density parity-check code are presented.

The logarithmic error (13) obviously depends on the input distribution p_Y. However, it is easily seen that the worst case distribution is one with $p_Y(y_j) = 1$ for some j. Only such worst case inputs are considered in the sequel. The distribution of (13) does then not depend on the size n of the input alphabet $\mathcal{Y} = \{y_1,\ldots,y_n\}$. Fig. 4 shows the standard deviation of the logarithmic error (13) as a function of the standard deviation of the transistor parameters ε, as obtained by numerical simulations.

We have also studied the effects of transistor mismatch for a binary (44,22,8) low-density parity-check code. Both the code and its decoder are defined by the factor graph shown in Fig. 5. (Factor graphs are defined in [4].) The code was not chosen for its performance, but for its suitability for studying various effects in both iterative digital and analog decoders.

Simulation results using (12) are shown in Fig. 6. All these simulations are for discrete-time sum-product decoders, but include transistor mismatch in the form (12). The simulation results

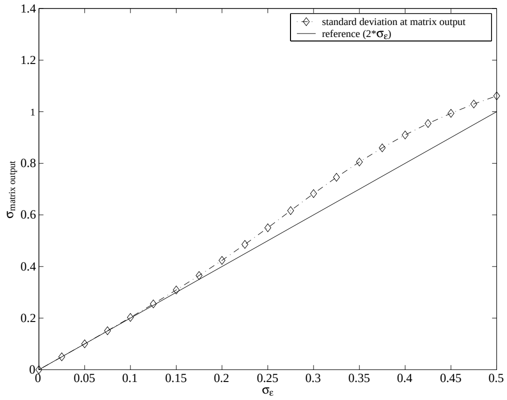

Figure 4: Standard deviation of the logarithmic error (13).

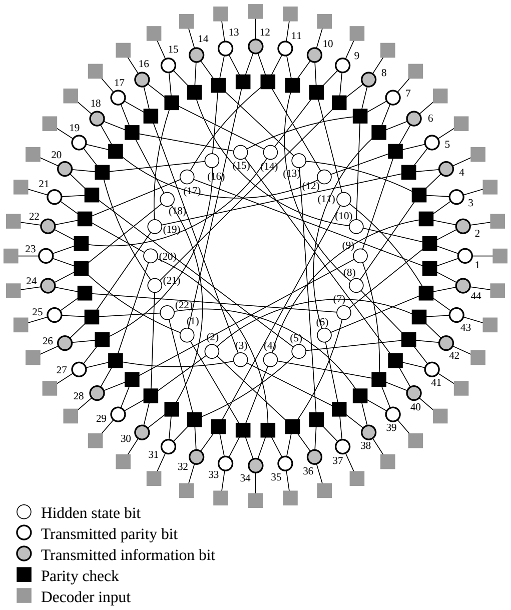

Figure 5: Factor-graph of the simulated code.

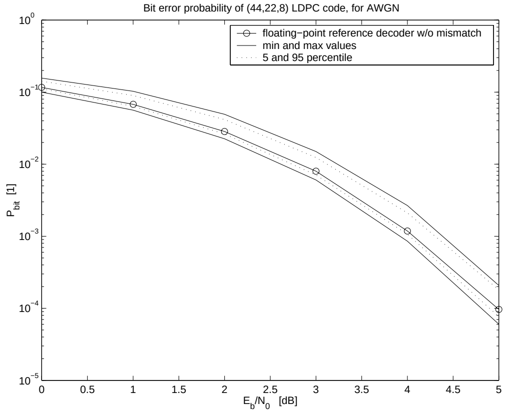

Figure 6: Simulation results for the (44,22,8) code with standard deviation of ε set to 10%.

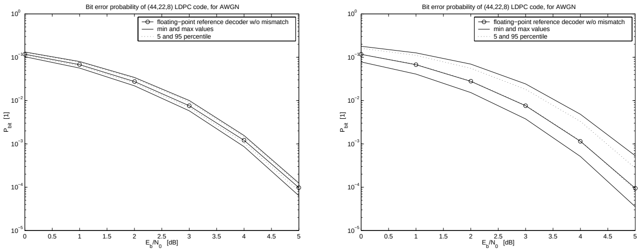

Figure 7: Simulation results for the (44,22,8) code with the standard deviation of ε set to 5% (on the left) and 15% (on the right).

for continuous-time versions are very similar. With an assumed standard deviation of 10 % for the mismatch parameters ε — a realistic value for relatively small transistors — 600 different decoding networks were generated. For each of these decoding networks, complete bit error rate (BER) curves were obtained by simulation. A second set of simulations for an assumed standard deviation of 5 % and 15 % for the mismatch parameters ε are shown in Fig. 7.

The top and the bottom solid lines in Fig. 6 and Fig. 7 show the worst and the best, respectively, of the simulated decoding networks. The dotted lines show the percentiles 5 and 95 of these decoding networks. Also shown by a solid line is the reference decoder without transistor mismatch, which is a standard discrete-time sum-product decoder. Surprisingly, the best decoding networks with random transistor mismatch outperform the reference decoder. This is unexpected, but not logically impossible: on graphs with interacting cycles, there are no optimality claims for the standard sum-product decoder.

7. CONCLUDING REMARKS

The accuracy of analog decoding circuits appears to be limited primarily by transistor mismatch. The analytical results of this paper allow to study such effects by fast high-level simulations.

The simulation software is also capable of handling further nonidealities such as propagation delays and MOS transistor characteristics, but we have not yet began to explore these issues systematically.

8. ACKNOWLEDGEMENT

The authors are indepted to Mr. Pascal Vontobel for contributing the code of Fig. 5 and assistance with coding issues, to Mr. Ermanno Schinca for substantially contributing to the software development, and to Mr. Stefan Moser for helpful discussions.

9. REFERENCES

[1] C. Berrou, A. Glavieux, and P. Thitimajshima. “Near Shannon-limit error-correcting coding and decoding: turbo codes”. In *Proc. ICC*, pp. 1064–1070. Geneva, May 1993.

[2] R. G. Gallager. *Low-Density Parity-Check Codes*. MIT Press, 1963.

[3] D. J. C. MacKay. “Good error-correcting codes based on very sparse matrices”. *IEEE Trans. Inf. Theory*, vol. 45, no. 2, pp. 399–431, March 1999.

[4] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger. “Factor graphs and the sum-product algorithm”, June 2000. To appear in *IEEE Trans. Inform. Theory*. Available at `http://www.comm.utoronto.ca/frank/factor`.

[5] G. D. Forney, Jr. “Codes on graphs: News and views”. In *Proc. Int. Symp. on Turbo Codes and Related Topics*, pp. 9–16. Brest, France, Sept. 2000.

[6] G. D. Forney, Jr. “On iterative decoding and the two-way algorithm”. In *Proc. Int. Symp. on Turbo Codes and Related Topics*. Brest, France, Sept. 1997.

[7] H.-A. Loeliger, M. Helfenstein, F. Lustenberger, and F. Tarköy. “Probability propagation and decoding in analog VLSI”. In *Proc. ISIT*, p. 146. Cambridge, MA, Aug. 1998.

[8] F. Lustenberger, M. Helfenstein, H.-A. Loeliger, F. Tarköy, and G. S. Moschytz. “An analog decoding technique for digital codes”. In *Proc. ISCAS*, vol. II, pp. 428–431. Orlando, FL, June 1999.

[9] H.-A. Loeliger, F. Lustenberger, M. Helfenstein, and F. Tarköy. “Probability propagation and decoding in analog VLSI”, Sept. 2000. To appear in *IEEE Transactions on Information Theory*, available at `http://www.isi.ee.ethz.ch/~lustenbe`.

[10] F. Lustenberger, M. Helfenstein, H.-A. Loeliger, F. Tarköy, and G. S. Moschytz. “All-analog decoder for a binary (18,9,5) tail-biting trellis code”. In *Proc. ESSCIRC*, pp. 362–365. Duisburg, Sep. 1999.

[11] F. Lustenberger. *On the Design of Analog VLSI Iterative Decoders*. Ph.D. thesis, Swiss Federal Institute of Technology, Zurich, November 2000.

[12] J. Hagenauer. “Decoding of binary codes with analog networks”. In *Proc. 1998 Information Theory Workshop*, pp. 13–14. San Diego, CA, Feb. 1998.

[13] M. Moerz, T. Gabara, R. Yan, and J. Hagenauer. “An analog 0.25μV BiCMOS tailbiting MAP decoder”. In *Proc. ISSCC*, pp. 356–357. San Francisco, CA, Feb. 2000.

[14] J. Hagenauer, M. Moerz, and E. Offer. “Analog turbo-networks in VLSI: the next step in turbo decoding and equalization”. In *Proc. Int. Symp. on Turbo Codes and Related Topics*, pp. 209–218. Brest, France, Sept. 2000.

[15] M. H. Shakiba, D. A. Johns, and K. W. Martin. “An integrated 200-MHz 3.3-V BiCMOS class-IV partial-response analog Viterbi decoder”. *IEEE J. Solid-State Circuits*, vol. 33, no. 1, pp. 61–75, Jan. 1998.

[16] M. H. Shakiba, D. A. Johns, and K. W. Martin. “BiCMOS circuits for analog viterbi decoders”. *IEEE Trans. CAS–II*, vol. 45, no. 12, pp. 1527–1537, Dec. 1998.

[17] A. Demosthenous and J. Taylor. “Low-power CMOS and BiCMOS circuits for analog convolutional decoders”. *IEEE Trans. CAS–II*, vol. 46, no. 8, pp. 1077–1080, Aug. 1999.

[18] K. He and G. Cauwenberghs. “An area-efficient analog VLSI architecture for state-parallel Viterbi decoding”. In *Proc. ISCAS*, vol. II, pp. 432–435. Orlando, Florida, May 1999.

[19] M. J. Pelgrom, A. C. J. Duinmaijer, and A. P. G. Welbers. “Matching properties of MOS transistors”. *IEEE J. Solid-State Circuits*, vol. 24, no. 5, pp. 1433–1439, Oct. 1989.