or

$$\Delta F_{TLS} = (E_1 + \Delta E_1 + \Gamma_1)^+ ((\Delta E_2 - \Delta E_1 F_c) + (\Gamma_2 - \Gamma_1 F_c)).$$

We note that the range space of Γ_1 and the range space of Γ_2 are orthogonal to the range space of $(E_1 + \Delta E_1 + \Gamma_1)$ [17], and hence,

$$\Delta F_{TLS} = (E_1 + \Delta E_1 + \Gamma_1)^+ (\Delta E_2 - \Delta E_1 F_c).$$

Using the fact that $\|\Gamma_1\|_F^2 \leq \|\Gamma_1, \Gamma_2\|_F^2 \leq \|\Delta E_1, \Delta E_2\|_F^2$ and using (A.3), as before, we obtain an approximation for ΔF_{TLS} which results in an expression for the mean squared error of the DOA estimate that is accurate to $o(N^{-1})$. The expression is exactly the same as that obtained for the LS approach, i.e., (A.4).

ACKNOWLEDGMENT

The authors would like to thank Prof. K. S. Arun of the University of Illinois, Urbana-Champaign, who pointed out the mistake in our 4th ASSP Spectrum Estimation Workshop paper, and for the derivation of (16). The first author would like to acknowledge a reviewer of [12] who also suggested the derivation of (16).

REFERENCES

- [1] R. Roy, A. Paulraj, and T. Kailath, "Estimation of signal parameters via rotational invariance techniques—ESPRIT," in *Proc. 19th Asilomar Conf. Circuits, Syst., Computing*, Asilomar, CA, Nov. 1985.
- [2] R. Kumaresan and D. W. Tufts, "Estimating the angle of arrival of multiple plane waves," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-19, Jan. 1983.
- [3] S. Y. Kung, C. K. Lo, and R. Foka, "A Toeplitz approximation approach to coherent source direction finding," in *Proc. ICASSP*, 1986.
- [4] R. O. Schmidt, "Multiple emitter location and signal parameter estimation," in *Proc. RADAR Spectral Estimation Workshop*, Griffiss AFB, NY, 1979.
- [5] B. D. Rao and K. V. S. Hari, "Performance analysis of Root-MUSIC," *IEEE Trans. Acoust., Speech, Signal Processing*, this issue, pp. 1939–1949.
- [6] M. Kaveh and A. J. Barabell, "The statistical performance of MUSIC and the minimum norm algorithms in resolving plane waves in noise," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-34, pp. 331–340, Apr. 1986.
- [7] D. J. Jeffries and D. R. Farrier, "Asymptotic results for eigenvector methods," *IEE Proc.*, vol. 132, pt. F, no. 7, pp. 589–594, Dec. 1985.
- [8] R. Roy and T. Kailath, "Total least-squares ESPRIT," in *Proc. 21st Asilomar Conf. Circuits, Syst., Computing*, Asilomar, CA, Nov. 1987.
- [9] R. Roy, A. Paulraj, and T. Kailath, "Comparative performance of ESPRIT and MUSIC for direction-of-arrival estimation," in *Proc. 20th Asilomar Conf. Circuits, Syst., Computing*, Asilomar, CA, Nov. 1986.
- [10] D. R. Brillinger, *Time Series: Data Analysis and Theory*. New York: Holt, Rinehart and Winston, 1975.
- [11] S. Y. Kung, K. S. Arun, and D. V. Bhaskar Rao, "State space and SVD based approximation methods for the harmonic retrieval problem," *J. Opt. Soc.*, vol. 73, pp. 1799–1811, Dec. 1983.
- [12] B. D. Rao, "Sensitivity considerations in state-space model-based harmonic retrieval methods," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, pp. 1789–1794, Nov. 1989.
- [13] —, "Sensitivity analysis of state space methods in spectrum estimation," in *Proc. ICASSP 87*, Dallas, TX, Apr. 6–9, 1987.
- [14] R. Roy, A. Paulraj, and T. Kailath, "ESPRIT—A subspace rotation approach to estimating parameters of cisoids in noise," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-34, pp. 1340–1342, Oct. 1986.
- [15] R. E. Skelton and D. A. Wagie, "Minimal root sensitivity in linear systems," *J. Guidance*, pp. 570–574, Sept. 1984.
- [16] G. W. Stewart, "The perturbation of the pseudo-inverses, projections, and linear least squares problems," *Siam Rev.*, vol. 19, no. 4, pp. 634–662, Oct. 1977.
- [17] G. H. Golub and C. F. Van Loan, *Matrix Computations*. Baltimore, MD: The Johns Hopkins University Press, 1983.

Matrix Based Computation of Floating-Point Roundoff Noise

BARRY D. VAN VEEN AND RICHARD BARANIUK

Abstract—A matrix based procedure is presented for computing the output roundoff noise power for filters implemented with floating-point arithmetic. The filter's computational structure is represented in terms of a product of matrices, known as a factored state variable description. The quantities needed to compute the output roundoff noise power are obtained from the factored state variable description via matrix manipulation. The expression for output roundoff noise power is shown to be of the same form as that for fixed-point arithmetic roundoff noise. Comparison indicates that, under very general conditions, fixed-point arithmetic provides better roundoff noise performance than floating point. Several examples are provided.

I. INTRODUCTION

The error due to roundoff in digital filters realized with floating-point arithmetic has been studied by several investigators, including [1]–[4]. Computation of the noise power at the output resulting from rounding of sums and products has been approached in the time domain for first- and second-order filters [2] and in the frequency domain for higher order filters [3], [4]. However, neither of these approaches is particularly amenable to computer implementation for arbitrary filter structures. In this correspondence, we describe a matrix based method for computing the roundoff noise power at the output of an arbitrary filter structure. It is derived from a factored state variable description [6] (FSVD) for the filter. We also show that, under very general conditions, fixed-point arithmetic provides better roundoff noise performance than floating-point arithmetic.

Section II reviews the models used for floating-point roundoff error and gives a brief introduction to FSVD's. The expression for roundoff noise power at the filter output is developed in Section III in terms of the FSVD, and is compared to the fixed-point arithmetic case. Two examples are given in Section IV, and a summary is provided in Section V.

II. BACKGROUND

A. Roundoff Noise Model

Roundoff errors occur for both multiplication and addition in floating-point arithmetic. Let $[\bullet]_q$ denote the result of applying rounding to the mantissa of a floating-point number, and let $z(k) = x(k) + y(k)$ or $z(k) = ax(k)$ represent the result of addition or multiplication. We have [5]

$$[z(k)]_q = z(k) + \epsilon(k) z(k) \quad (1)$$

Manuscript received September 11, 1988; revised February 26, 1989.

B. D. Van Veen is with the Department of Electrical and Computer Engineering, University of Wisconsin, 1415 Johnson Drive, Madison, WI 53706.

R. Baraniuk is with the Department of Electrical and Computer Engineering, University of Illinois, 1406 West Green Street, Urbana, IL 61801. IEEE Log Number 8931335.

where $-q/2 \leq \epsilon(k) \leq q/2$ for a quantization interval of q . Assuming $\epsilon(k)$ is a white noise sequence uncorrelated with $z(k)$ and uniformly distributed on $-q/2$ to $q/2$, it follows that the roundoff error $e(k) = \epsilon(k)z(k)$ is white with variance $\sigma_e^2 = \sigma_z^2 E\{z^2(k)\}$ where $\sigma_e^2 = q^2/12$ [5].

Let $e_j(k)$ and $e_j(k)$ be associated with the j th arithmetic operation in the filter. Assuming that $e_j(k)$ is uncorrelated with $e_i(k)$ for all $i \neq j$, and is uncorrelated with all other variables in the filter, then the total output power due to roundoff error is the sum of the output powers due to each error source $e_j(k)$. Let $g_j(k)$ denote the unit pulse response from the node at which the j th error occurs to the output. Since $e_j(k)$ is a white sequence, it follows that the total output power due to roundoff error (σ_y^2) is given by

$$\sigma_y^2 = \sum_{j=1}^M \left(\sigma_{e_j}^2 \sum_{k=0}^{\infty} g_j^2(k) \right) \quad (2)$$

where M represents the total number of addition and multiplication operations in the filter. Evaluation of σ_y^2 requires computation of the norm of the unit pulse responses from each node to the output and the variance of the variable at each node, $\sigma_{e_j}^2$.

B. Factored State Variable Descriptions

Let the state variable description of a filter be given as

$$\begin{bmatrix} x(k+1) \\ y(k) \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} x(k) \\ u(k) \end{bmatrix} \quad (3)$$

where $x(k)$ is the n -dimensional state vector at time k , $u(k)$ is the input, and $y(k)$ is the output. Assuming $u(k)$ and $y(k)$ are scalars, then A is n by n , B is n by 1, C is 1 by n , and D is 1 by 1. The state variable description does not represent the filter in enough detail for our purpose. An FSVD provides a more detailed matrix based description of the filter structure. In terms of an FSVD, (3) is written

$$\begin{bmatrix} x(k+1) \\ y(k) \end{bmatrix} = F_L \cdots F_2 F_1 \begin{bmatrix} x(k) \\ u(k) \end{bmatrix}. \quad (4)$$

The F_i represent each stage of the internal filter computation and implicitly define a vector sequence of variables $q_i = F_i q_{i-1}$, $1 \leq i \leq L$, with $q_0 = [x^T(k) u(k)]^T$. Each element of q_i represents the result of the operation performed by the corresponding row of F_i . Denote the dimension of q_i as n_i . Note that the F_i must only be conformable and are not required to be of the same dimension. A more detailed discussion of FSVD's including procedures for their construction is given in [6].

III. COMPUTATION OF OUTPUT ROUND OFF NOISE POWER FROM THE FSVD

First construct an FSVD which represents each addition and multiplication in the filter as a separate operation. This implies that each row of each F_i will contain two unity elements amongst all zeros, representing an addition, or a single nonzero element, representing multiplication (unity elements correspond to trivial multiplications and do not produce any roundoff noise). Evaluation of $\sigma_{e_j}^2$ in (2) requires computation of $E\{z_j^2(k)\}$. Each $z_j(k)$ corresponds to an element of one of the q_i in the FSVD. Let $K_i = E\{q_i q_i^T\}$ be the n_i by n_i covariance of q_i (we assume $u(k)$ is zero mean). The diagonal elements of the K_i correspond to the variance of the internal filter variables $z_j(k)$. Given K_0 , it is straightforward to obtain K_i , $1 \leq i \leq L$ using

$$K_i = F_i K_{i-1} F_i^T \quad 1 \leq i \leq L. \quad (5)$$

K_0 is a function of A , B , and the input. Partition K_0 as

$$K_0 = \begin{bmatrix} \tilde{K} & f \\ f^T & r_u(0) \end{bmatrix} = \begin{bmatrix} E\{x(k)x^T(k)\} & E\{x(k)u(k)\} \\ E\{u(k)x^T(k)\} & E\{u^2(k)\} \end{bmatrix} \quad (6)$$

and define $r_u(k)$ as the autocorrelation sequence of the input. $x(k)$ can be expressed in terms of the input as

$$x(k) = \sum_{p=0}^{\infty} A^p B u(k-p-1) \quad (7)$$

which implies that

$$f = \sum_{p=0}^{\infty} A^p B r_u(p+1) \quad (8)$$

and

$$\tilde{K} = \sum_{m=0}^{\infty} \sum_{p=0}^{\infty} A^p B r_u(p-m) (A^m B)^T. \quad (9)$$

If $u(k)$ is unit variance white noise, we have $f = 0$ and $\tilde{K}$ satisfies [6, p. 318]

$$\tilde{K} = A \tilde{K} A^T + B B^T. \quad (10)$$

Using the identity $\text{vec}(SUV) = (V^T \otimes S) \text{vec}(U)$ [7] where $\text{vec}(\cdot)$ is the column stacking operator and $\otimes$ denotes the Kronecker product, it is easy to show that $\tilde{K}$ in (10) satisfies the set of linear equations

$$(I - A \otimes A) \text{vec}(\tilde{K}) = \text{vec}(B B^T). \quad (11)$$

Alternatively, a simple iterative algorithm for computing $\tilde{K}$ satisfying (10) is given in [6, p. 392].

We now focus attention on computation of the energy in the unit pulse responses. The vector pulse response from nodes associated with state variables to the output, $g(k)$, is given in terms of A and C as

$$g^T(k) = \begin{cases} 0, & k \leq 0 \\ C A^{k-1}, & k > 0. \end{cases} \quad (12)$$

Define the matrix $\tilde{W}$ as

$$\tilde{W} = \sum_{k=0}^{\infty} g(k) g^T(k). \quad (13)$$

The diagonal elements of $\tilde{W}$ represent the energies in the unit pulse responses from the state variable nodes to the output. Substituting (12) into (13) and comparing to (9) for white noise inputs provides the dual of (10)

$$\tilde{W} = A^T \tilde{W} A + C^T C. \quad (14)$$

The energies in the unit pulse responses from the nodes associated with the elements of the q_i to the output are obtained as the diagonal elements of n_i by n_i dimensional matrices W_i , $1 \leq i \leq L$, satisfying [6]

$$W_{i-1} = F_i^T W_i F_i, \quad 1 < i \leq L \quad (15)$$

where

$$W_L = \begin{bmatrix} \tilde{W} & 0 \\ 0 & 1 \end{bmatrix}.$$

Note that (11) (or the algorithm in [6]) can be used to compute $\tilde{W}$ by replacing A with A^T and B with C^T .

The quantities required for computing the output roundoff noise power are now available. Define indicator variables γ_{ij} , $1 \leq i \leq L$, $1 \leq j \leq n_i$, so that $\gamma_{ij} = 0$ if the j th row of F_i contains a single unity element and $\gamma_{ij} = 1$ otherwise. Variables at which rounding occurs have $\gamma_{ij} = 1$. The output roundoff noise power is expressed as

$$\sigma_y^2 = \frac{q^2}{12} \sum_{i=1}^L \sum_{j=1}^{n_i} \gamma_{ij} [K_i]_{jj} [W_i]_{jj} = \frac{q^2}{12} N G_{ff} \quad \text{floating point} \quad (16)$$

where $[Q]_{ii}$ indicates the i th diagonal element of Q .

A similar expression is derived for the output roundoff noise power in fixed-point arithmetic in [6]. In fixed-point arithmetic, roundoff noise is only generated by multiplication, and the internal filter variables must be scaled to prevent overflow. The diagonal elements of K_i represent the variance of the internal filter variables and provide the information needed for L_2 scaling. The roundoff noise at each node is amplified by the energy in the unit pulse response from that node to the output, obtained from the diagonal elements of the W_i (see [6] for a more detailed discussion). Assuming single length accumulators, the output roundoff noise power is given by

$$\sigma_y^2 = \frac{q^2 \delta^2}{12} \sum_{i=1}^L \sum_{j=1}^{n_i} \beta_{ij} [K_i]_{jj} [W_i]_{jj} = \frac{q^2 \delta^2}{12} NG_{fx} \quad (\text{fixed point}) \quad (17)$$

where $\beta_{ij} = 1$ if the j th row of F_i contains a multiplication by a number other than 1 and $\beta_{ij} = 0$ otherwise. q represents the quantization interval and δ is the scaling parameter which specifies how many standard deviations an internal variable must exceed before an overflow occurs. If the input is unit variance white noise, then the variance of the internal variables is δ^{-2} after scaling. Typically, $3 \leq \delta \leq 5$, with larger values representing more conservative scaling strategies.

Equations (16) and (17) are almost identical except for δ^2 and the number of terms included by the indicator variables. In floating-point arithmetic, the exponent effectively scales variables to prevent overflow automatically and K_i arises because the roundoff noise power is proportional to variable variance. In contrast, the roundoff noise power contributed by each variable is the same in fixed-point arithmetic and K_i is a result of scaling internal variables manually.

The comparison of roundoff error powers for fixed- and floating-point arithmetic depends on δ , the quantization intervals, and the roundoff error power contributed by terms corresponding to additions in (16). Note that $NG_{fx} < NG_{fl}$ because NG_{fx} excludes terms due to additions. (NG_{fx} can be made even smaller by using double length accumulators.) Assume the same number of bits are used in floating- and fixed-point arithmetic and that b_e bits are employed for the floating-point exponent. This implies that $q_{fl} = 2^{b_e} q_{fx}$. The ratio of floating-point to fixed-point output roundoff noise power is expressed as

$$r = \frac{4^{b_e} NG_{fl}}{\delta^2 NG_{fx}} \quad (18)$$

r is guaranteed to be greater than one if $4^{b_e} \geq \delta^2$. For example, if $\delta = 5$ and $b_e = 4$ (three bit exponent plus a sign bit), then the floating-point roundoff noise power is at least a factor of ten greater than fixed point. Equation (18) indicates fixed-point arithmetic employed in a manually scaled filter makes more efficient use of a given number of bits than floating-point arithmetic where some bits are allocated to an exponent for automatic scaling.

Recall that the floating-point roundoff noise is proportional to input signal level. Thus, the floating-point system maintains an approximately constant signal-to-roundoff noise ratio at the output. In contrast, the fixed-point roundoff noise is independent of the input signal level and the output roundoff noise power is constant.

IV. EXAMPLES

The first example is the second-order direct form filter considered in [1] and [3]. The transfer function is

$$\frac{Y(z)}{U(z)} = \frac{1}{1 + a_1 z^{-1} + a_2 z^{-2}} \quad (19)$$

where $a_1 = -2^{1/2} (1 - 0.001)$ and $a_2 = (1 - 0.001)^2$. The signal flow graph (SFG) in Fig. 1 depicts the filter and sources of roundoff

Fig. 1. Signal flow graph for second-order filter considered in [1] and [3]. Sources of roundoff error are indicated by e_j . Unlabeled branches have unit gain.

Fig. 2. Signal flow graph for a second-order section in sixth-order filter. Sources of roundoff error are indicated by e_j . Unlabeled branches have unit gain.

error. The FSVD corresponding to this SFG is given by

$$F_1 = \begin{bmatrix} 0 & 0 & 1 \\ a_2 & 0 & 0 \\ 0 & a_1 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad F_2 = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (20)$$

$$F_3 = \begin{bmatrix} 0 & 0 & 1 \\ -1 & 1 & 0 \end{bmatrix}, \quad F_4 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}.$$

Roundoff noise is generated at the nodes corresponding to the second and third rows of F_1 , the first row of F_2 , and the second row of F_3 . Assuming unit variance white noise input to the filter, we obtain $NG_{fl} = 1.25 \times 10^6$, which is consistent with the result in [3]. In fixed-point arithmetic, we obtain $NG_{fx} = 7.5 \times 10^5$. Thus, the noise gain due to additions in floating-point arithmetic is 5×10^6 and $r = 1.67(4^{b_e})/\delta^2$.

The second example is a sixth-order Butterworth low-pass filter with cutoff frequency $\pi/16$ rad/s. The filter is implemented as a cascade of second-order sections. Each second-order section is optimized with respect to fixed-point arithmetic roundoff noise using the block optimal design procedure in [6, pp. 400–402]. The SFG for a section is depicted in Fig. 2 with sources of roundoff error indicated. The FSVD for the cascade is obtained in terms of the FSVD for each section. Thus, the FSVD is given by a product of nine matrices. Evaluation of (16) gives $NG_{fl} = 2.835$ when the

input is unit variance white noise. In fixed-point arithmetic, we obtain $NG_{fx} = 0.879$. In this case we have $r = 3.2(4^b)/\delta^2$.

V. SUMMARY

FSVD's are employed to analyze the output roundoff noise power of arbitrary filter structures implemented in floating-point arithmetic. An expression is developed for the output roundoff noise power which is very similar in form to that for fixed-point arithmetic. This expression involves matrix operations and is easily evaluated on a computer. A comparison of fixed- and floating-point arithmetic indicates that, for very general conditions, fixed-point arithmetic provides better roundoff noise performance. This is because bits are used more efficiently by manually scaling the filter in fixed-point arithmetic than by allocating bits to an exponent for automatic scaling as occurs in floating-point arithmetic. Two examples are provided.

REFERENCES

- [1] I. W. Sandberg, "Floating point roundoff accumulation in digital filter realizations," *Bell Syst. Tech. J.*, vol. 46, pp. 1775-1791, Oct. 1967.
- [2] C. Weinstein and A. V. Oppenheim, "A comparison of roundoff noise in floating point and fixed point digital filter realizations," *Proc. IEEE*, vol. 57, pp. 1181-1183, June 1969.
- [3] B. Liu and T. Kaneko, "Error analysis of digital filters realized with floating point arithmetic," *Proc. IEEE*, vol. 57, pp. 1735-1747, Oct. 1969.
- [4] E. P. F. Kan and J. K. Aggarwal, "Error analysis of digital filter employing floating point arithmetic," *IEEE Trans. Circuit Theory*, vol. CT-18, pp. 678-686, Nov. 1971.
- [5] A. V. Oppenheim and R. W. Schaffer, *Digital Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1975.
- [6] R. A. Roberts and C. T. Mullis, *Digital Signal Processing*. Reading, MA: Addison-Wesley, 1987.
- [7] J. W. Brewer, "Kronecker products and matrix calculus in system theory," *IEEE Trans. Circuits Syst.*, vol. CAS-25, pp. 772-781, Sept. 1978.