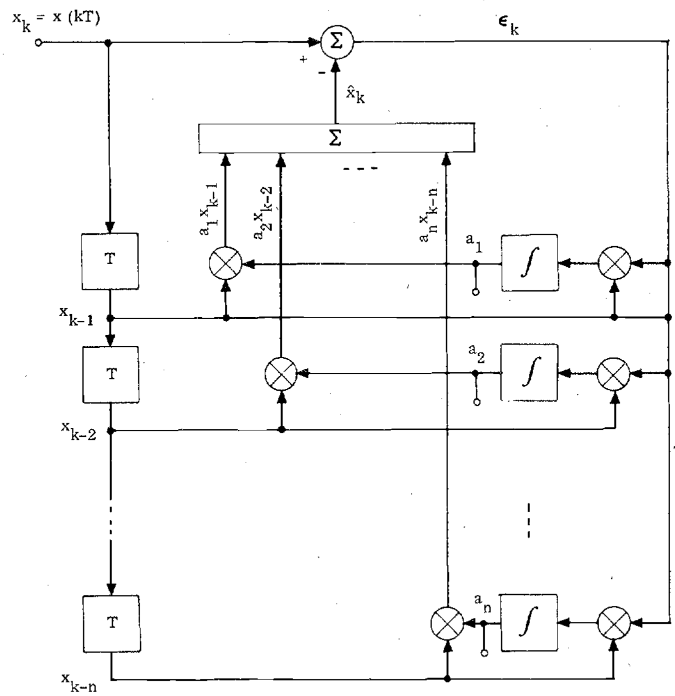

Fig. 2. Adaptive linear predictor using CCL's.

drawn. In particular, problems of coefficient estimate variance [23], [30] and stability [1], [17] need to be investigated in detail.

REFERENCES

[1] B. S. Atal and S. L. Hanauer, "Speech analysis and synthesis by linear prediction of the speech wave," *J. Acoust. Soc. Amer.*, vol. 50, no. 2, part 2, pp. 637–655, 1971.
[2] J. D. Markel, "Formant trajectory estimation from a linear least-squares inverse filter formulation," Speech Communication Research Lab Inc., Santa Barbara, Calif., AD 734679, Oct. 1971.
[3] J. D. Markel, "Digital inverse filtering—A new tool for formant trajectory estimation," *IEEE Trans. Audio Electroacoust.*, vol. AU-20, pp. 129–137, June 1972.
[4] R. G. Crichton and F. Fallside, "Speech analysis-synthesis on a small computer," in *Proc. Spring Meeting British Acoust. Soc.*, vol. 2, Apr. 1973, paper 73SHB7.
[5] J. Rogers, "Digital inverse filtering of the speech waveform," in *Proc. Spring Meeting British Acoust. Soc.*, vol. 2, Apr. 1973, paper 73SHB6.
[6] J. D. Markel and A. H. Gray, Jr., "On autocorrelation equations as applied to speech analysis," *IEEE Trans. Audio Electroacoust.*, vol. AU-21, pp. 69–79, Apr. 1973.
[7] J. N. Maksym, "Real-time pitch extraction by adaptive prediction of the speech waveform," *IEEE Trans. Audio Electroacoust.*, vol. AU-21, pp. 149–154, June 1973.
[8] J. R. Haskew, J. M. Kelly, R. M. Kelly, Jr., and T. H. McKinney, "Results of a study of the linear prediction vocoder," *IEEE Trans. Commun.*, vol. COM-21, pp. 1008–1015, Sept. 1973.
[9] J. R. Welch and J. D. Oetting, "Formant extraction hardware using adaptive linear predictive coding," in *Proc. Nat. Telecommunications Conf.*, Nov. 1973, paper 29C.
[10] D. T. Magill, "Adaptive speech compression for packet communications systems," *Proc. Nat. Telecommunications Conf.*, Nov. 1973, paper 29D.
[11] C. Klayman *et al.*, "Real-time implementation of a linear predictive coding system," *Proc. Nat. Communications Conf.*, Nov. 1973, paper 29E.
[12] J. D. Markel and A. H. Gray, Jr., "A liner prediction vocoder simulation based upon the autocorrelation method," *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. ASSP-22, pp. 124–134, Apr. 1974.
[13] M. R. Schroeder, "Vocoders: Analysis and synthesis of speech," *Proc. IEEE*, vol. 54, pp. 720–734, May 1966.
[14] B. Gold and C. M. Rader, "The channel vocoder," *IEEE Trans. Audio Electroacoust.*, vol. AU-15, pp. 148–161, Dec. 1967.
[15] A. Papoulis, *Probability, Random Variables and Stochastic Processes*. New York: McGraw-Hill, 1965.
[16] J. E. Gunn and A. P. Sage, "Speech data rate reduction—Part I: Applicability of modern estimation theory," *IEEE Trans. Aerosp. Electron Syst.*, vol. AES-9, pp. 130–138, Mar. 1973.
[17] J. L. Melsa *et al.*, "Vocal tract parameter identification using sequential estimation techniques," in *Proc. Nat. Telecommunications Conf.*, Nov. 1973, paper 29A.
[18] F. Itakura and S. Saito, "Digital filtering techniques for speech analysis and synthesis," presented at the 7th Int. Conf. Acoustics, Budapest, Hungary, 1971, paper 25C1.
[19] F. Itakura, S. Saito, T. Koike, H. Sawabe, and M. Nishikawa, "An audio response unit based on partial autocorrelation," *IEEE Trans. Commun.*, vol. COM-20, pp. 792–797, Aug. 1972.
[20] J. G. Dunn *et al.*, "Progress in the development of a digital vocoder employing an Itakura adaptive predictor," in *Proc. Nat. Telecommunications Conf.*, Nov. 1973, paper 29B.
[21] P. W. Howells, "Intermediate frequency side-lobe canceller," U.S. Patent 3 202 990, Aug. 24, 1965.
[22] B. Widrow, P. E. Mantey, L. J. Griffiths, and B. B. Goode, "Adaptive antenna systems," *Proc. IEEE*, vol. 55, pp. 2143–2159, Dec. 1967.
[23] B. Widrow, "Adaptive filters I: Fundamentals," Stanford Electronics Labs., Stanford, Calif., Rep. SEL-66-126 (Tech. Rep. 6764-6), Dec. 1966.
[24] —, "Adaptive filters," in *Aspects of Network and System Theory*, part IV, R. E. Kalmen and N. DeClaris, Eds. New York: Holt, Rinehart and Winston, 1971, pp. 563–587.
[25] R. W. Lucky, "Adaptive redundancy removal in data transmission," *Bell Syst. Tech. J.*, pp. 549–573, Apr. 1968.
[26] D. R. Morgan, "An adaptive linear predictor," Electronics Laboratory, General Electric Co., internal memo 35-EL-1268, Jan. 1973.
[27] L. S. Moye, "Self adaptive filter," U.K. Patent 1 184 653, Mar. 18, 1970.
[28] —, "Digital transmission of speech at low bit rates," *Elec. Commun.*, vol. 47, no. 4, pp. 212–223, 1972.
[29] B. S. Atal and M. R. Schroeder, "Adaptive predictive coding of speech signals," *Bell Syst. Tech. J.*, vol. 49, pp. 1973–1986, Oct. 1970.
[30] L. E. Brennan, E. L. Pugh, and I. S. Reed, "Control-loop noise in adaptive array antennas," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-7, pp. 254–262, Mar. 1971.

A Note on Implementation of Digital Filters

A. PELED, B. LIU, AND K. STEIGLITZ

***Abstract*—A recently proposed method to implement digital filters using ROM can also be employed to implement multiplication by a constant. This note discusses alternate implementations for digital filter sections substituting memory for logic and points out the possible advantage of doing so.**

Recently, a method was proposed [1]–[3] to implement digital filters which computes the sum of products of constants times signal values by accumulating the results of table look-ups in a ROM. For the purposes of this note, we will call this the "coefficient slicing" (CS) implementation. The replacement of multiplier hardware by ROM hardware can lead to significant improvements in the time-hardware product in situations where the constants (filter coefficients) are limited to some fixed collection of values, say as many as 128 different sets of second-order filter coefficients.

The same approach as that employed in [1]–[3] can be used to implement multiplication by a constant chosen from a set of fixed constants, and it is the purpose of this note to point out the possible advantages of doing so. In particular, a second-order section with three multiplications (such as might be employed in a parallel realization) can be implemented by such a ROM multiplication by a constant (RMC) with efficiency comparable to the equivalent bit serial CS implementation. As described in [1], the implementation of such a CS section requires one accumulator, plus one $2^r * 16$-bit ROM for each set of r 16-bit coefficients. Thus, the hardware required is

$$H_{CS} = H_1 + n2^rH_2 \tag{1}$$

units, where n is the number of fixed sections allowed, H_2 is the amount of hardware corresponding to a $1 * 16$-bit ROM, and H_1 is the amount of fixed hardware. The unit of hardware is arbitrary, and may be IC's, gates, power, cost, or some combination. The time to produce an output word is $B \cdot t_\alpha$, where B is the number of bits in the data word, and t_α is the maximum of the cycle time of the ROM or adder.

Manuscript received March 6, 1974; revised January 2, 1975. This work was supported in part by the Air Force Office of Scientific Research, USAF, under Grant AF-AFOSR 71-2101, and in part by the U.S. Army Research Office, Durham, N.C., under Contract DA-HC04-69-C-0012.

The authors are with the Department of Electrical Engineering, Princeton University, Princeton, N.J. 08540.

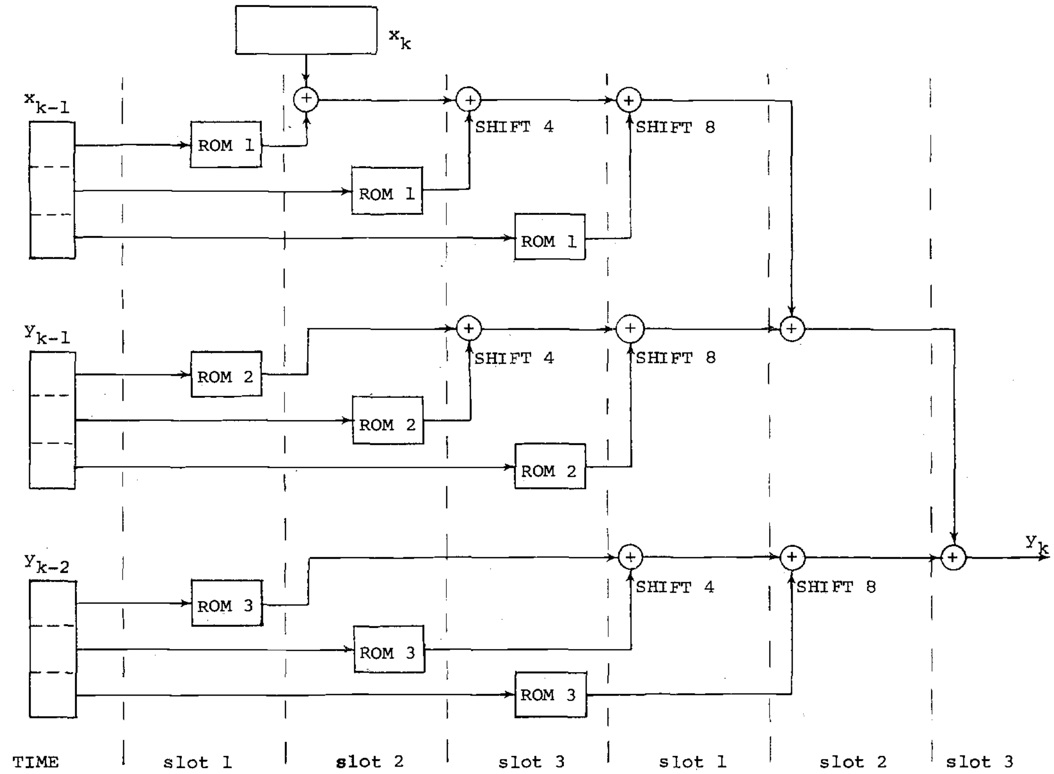

Fig. 1. Flow diagram for a ROM multiply by a constant RMC (12 bit data taken in groups of 4 bits).

Thus, the time-hardware product is

$$\mathrm{TH}_{\mathrm{CS}} = (H_1 + n2^r H_2) \cdot Bt_\alpha \tag{2}$$

which, incidently, has the units of energy-per-word if the hardware is evaluated in units of power.

Next, consider the multiplication of a constant c by a positive variable x. Break x into k parts of B/k bits each (assume k divides B), and write

$$cx = c \sum_{i=1}^{k} x_i 2^{-(i-1)B/k} = \sum_{i=1}^{k} (cx_i) 2^{-(i-1)B/k} \tag{3}$$

where x_i is a B/k-bit segment. From this, cx can be computed by accumulating k values of (cx_i), each of which can be obtained from a $2^{B/k} * 16$-bit ROM. If the variable x is a signed number then, depending upon the particular representation used, a minor modification of (3) is needed. For the 2's complement representation since the sign bit has to be interpreted as negative, in order to be able to use the same ROM for the sign bit and for the other bits, the sign bit has to be put in a group by itself, complementing it with $(B/k - 1)$ zeros. For more details of this case see [2]. For the sign-magnitude representation the entries in the ROM will be in sign-magnitude form too and the sign of the output will be determined by an additional table look-up addressed by the sign bit of x and by the index number of the constant c in the set of possible constant multipliers implemented. For the rest of this discussion we assume that sign-magnitude representation is used. Thus the time for such a multiplication is $k \cdot t_\alpha$ so the time-hardware product for the filter section is

$$\mathrm{TH}_{\mathrm{RMC}} = r(H_1 + 2^{B/k} nH_2) kt_\alpha \tag{4}$$

assuming we implement r parallel RMC's.

This is based on the observation that the H_1-part of both systems is roughly the same. The ratio of time-hardware products is therefore

$$\frac{\mathrm{TH}_{\mathrm{RMC}}}{\mathrm{TH}_{\mathrm{CS}}} = \frac{rk}{B} \left(\frac{\dfrac{H_1}{nH_2} + 2^{B/k}}{\dfrac{H_1}{nH_2} + 2^r} \right). \tag{5}$$

When $B/k = r$, this ratio is precisely 1; at this point the two systems are essentially equal in efficiency. When $B/k = r + 1$, the RMC implementation is about $1/(r + 1)$ more efficient, assuming $H_1/nH_2 >> 2^r$. If $H_1/nH_2 >> 2^{r+1}$, even more savings are possible. Loosely speaking, the higher B/k (bits addressed at once), the greater the efficiency, up to the point that the total ROM storage becomes excessive. We note here that the foregoing analysis assumes that the total traffic justifies the construction of total hardware in the amount given in (4): r RMC's. When this is not true, the advantages of the CS implementation become apparent.

Fig. 1 shows a flow diagram for the case of the second-order section corresponding to $r = 3$.

$$y_j = x_j + c_1 x_{j-1} + c_2 y_{j-1} + c_3 y_{j-2}$$

and the choices $B = 12$ and $k = 3$. Three parallel paths are implemented, and three cycles are needed to fetch the partial products and accumulate them. As can be seen, every adder and ROM is busy all the time, with some adders accumulating sums from previous words while their corresponding ROM's are fetching partial products for the present word.

The discussion above allows us to state things in terms of single multipliers, and to see roughly why and when it becomes advantageous because of large n (number of sections) to build a general purpose multiplier. Taking $B = 12$, $k = 3$ as above, an RMC corresponds to a multiplier with speed $3 \cdot t_\alpha$ and hardware $H_1 + 16nH_2$. With $t_\alpha = 35$ ns, this corresponds to a 105 ns, multiplier. If $H_1 = 10$ IC's (for an accumulator, registers, and control) and $H_2 = 1/256$ IC (assuming 512×8 bits/IC such as Signetics 8205), this means we can implement this 105 ns RMC with n constants using $10 + n/16$ IC's. Compared with an array multiplier using 18 IC's and operating at the same speed, RMC implementation would be advantageous up to $n = 128$. In a particular application (4) allows us to choose k optimally for an RMC, and this choice depends on n, the number of constants the multiplier must be able to provide.

The purpose of this note was to point out alternate implementations for digital filter sections substituting memory for logic. We thank the reviewers for pointing out that the RMC algorithm is well known, both for complete and partial product look-up. See, for example, [4].

References

[1] A. Peled and B. Liu, "New hardware realizations of digital filters," *IEEE Trans. Acoust., Speech, and Signal Processing*, vol. ASSP-22, pp. 456-462, Dec. 1974.

[2] B. Liu and A. Peled, "A new hardware realization of high speed FFT's," to be published.
[3] A. Croisier, D. J. Esteban, M. E. Levilion, and V. Rizo, "Digital filter for PCM encoded signals," U.S. Patent 3 777 130, Dec. 3, 1973.
[4] H. W. Kaiser *et al.*, "Digital filters for high speed data processing," RCA Tech. Rep. AFAL-TR-70-204, Sept. 1970.

Comments on "An l_p Design Technique for Two-Dimensional Digital Recursive Filters"

N. A. PENDERGRASS

Abstract—**It is shown that examples can be found for which the l_p design algorithm in two dimensions will attempt to converge to an unstable solution.**

An algorithm is proposed in the above-mentioned paper by Maria and Fahmy[1] for the design of two-dimensional recursive digital filters. They suggest that the algorithm assures convergence to a stable filter. This is justified by citing a paper by Deczky [1] in which a similar algorithm is described for the one-dimensional problem. However, in the two-dimensional case, counterexamples can be devised which show that the algorithm in the Maria and Fahmy paper may attempt to converge to an unstable solution. One such counterexample will be given in this correspondence.

The design approach in the above paper[1] is to choose the parameters of a two-dimensional digital transfer function

$$F(z_1, z_2) = A \frac{\prod_{l=1}^{K_1} N^{(l)}(z_1, z_2)}{\prod_{l=1}^{K_2} D^{(l)}(z_1, z_2)} \tag{1}$$

where

$$N^{(l)}(z_1, z_2) = [1\ z_1\ z_1^2] \begin{bmatrix} 1 & a_{12}^l & a_{13}^l \\ a_{21}^l & a_{22}^l & a_{23}^l \\ a_{31}^l & a_{32}^l & a_{33}^l \end{bmatrix} \begin{bmatrix} 1 \\ z_2 \\ z_2^2 \end{bmatrix}$$

$$D^{(l)}(z_1, z_2) = [1\ z_1\ z_1^2] \begin{bmatrix} 1 & d_{12}^l & d_{13}^l \\ d_{21}^l & d_{22}^l & d_{23}^l \\ d_{31}^l & d_{32}^l & d_{33}^l \end{bmatrix} \begin{bmatrix} 1 \\ z_2 \\ z_2^2 \end{bmatrix}$$

to minimize the performance index

$$J(\phi) = \sum_{m=1}^{M} \sum_{n=1}^{N} [|F_{mn}| - Y_{mn}]^p \tag{2}$$

where ϕ is the vector of parameters, p is a positive even integer, Y_{mn} is the desired magnitude response at (ω_m, ω_n), and $|F_{mn}|$ is the actual magnitude response at (ω_m, ω_n). The algorithm proposed by Maria and Fahmy calculates each step $\Delta\phi$ by the Newton method. The new filter with parameters $\phi + \Delta\phi$ is then checked for stability, and if found unstable, that portion of $\Delta\phi$ corresponding to the unstable section is successively multiplied by $\frac{1}{2}$ until a stable filter is achieved. From this point, a new $\Delta\phi$ is computed and the process is continued until some criterion is met.

Manuscript received July 29, 1974; revised August 22, 1974 and January 15, 1975. This work was supported by the National Science Foundation under Grant GK 38205.

The author is with the Department of Electrical Engineering and Computer Science and the Electronics Research Laboratory, University of California, Berkeley, Calif. 94720.

[1] G. A. Maria and M. M. Fahmy, *IEEE Trans. Acoust., Speech, and Signal Processing*, vol. ASSP-22, pp. 15-21, Feb. 1974.

For a one-dimensional problem, it is shown in [1] that the performance index will always increase as ϕ is chosen closer and closer to the point where it produces an unstable filter. Therefore, from a stable point, there exists in all directions where $J(\phi)$ is decreasing a certain magnitude $\Delta\phi$ below which the algorithm will remain in the stable region. The two-dimensional performance index, however, does not always have this property, as we will next show.

Let us assume that the numerator and denominator of (1) have no common factors. We can easily show by example that $F(z_1, z_2)$ can be unstable with singular points on the unit sphere while having finite magnitude there. Note that this cannot occur in one dimension since we have assumed that common factors do not exist in the numerator and denominator.

Consider, for example,

$$F(z_1, z_2) = \frac{(z_1 + 1)(z_2 + 1)}{1 + \frac{1}{2} z_1 + \frac{1}{2} z_2}. \tag{3}$$

Setting the denominator equal to zero, we get

$$z_1 + z_2 = -2. \tag{4}$$

This curve crosses the unit sphere only at $(e^{\pm j\pi}, e^{\pm j\pi})$. While the frequency response $F(e^{j\theta_1}, e^{j\theta_2})$ is not continuous in θ_1 and θ_2 at these points, the limit along any path on the unit sphere is bounded.[2]

Furthermore, we can start with a stable filter and proceed to this unstable filter without necessarily increasing the performance index. Consider the filter

$$F(z_1, z_2) = A \left(\frac{1 + a_{10} z_1 + a_{01} z_2 + a_{11} z_1 z_2}{1 + b_{10} z_1 + b_{01} z_2 + b_{11} z_1 z_2} \right) \tag{5}$$

and consider a path given by

$$\left. \begin{aligned} A &= 1 \\ a_{10} &= a_{01} = a_{11} = 1 \\ b_{11} &= 0 \\ b_{10} &= b_{01} = \tau \end{aligned} \right\} \tag{6}$$

where τ starts at some value $0 < \tau < \frac{1}{2}$. The stability criterion along this path is [2] given by $|b_{10}| + |b_{01}| < 1$. Therefore, the starting point and all points along the path are stable short of $\tau = \frac{1}{2}$. Evaluated on the unit sphere, the frequency response as a function of τ is given by

$$F(e^{j\theta_1}, e^{j\theta_2}, \tau) = \frac{(e^{j\theta_1} + 1)(e^{j\theta_2} + 1)}{1 + \tau(e^{j\theta_1} + e^{j\theta_2})}.$$

If we choose $Y_{mn} = |F_{mn}|$ in (2) from this equation for $\tau = \frac{1}{2}$, then we can obviously find a stable starting point τ near $\frac{1}{2}$ from which $J(\phi)$ will decrease as τ goes to $\frac{1}{2}$.

This example, while very simple for convenience, shows that some paths along which $J(\phi)$ decreases may not stay inside the stable region. Hence, the existence of a stable local minimum of $J(\phi)$ is not guaranteed from an arbitrary starting point.

Let us consider the computation aspects of this example. For $\tau = \frac{1}{2}$, computation of $|F(e^{j\theta_1}, e^{j\theta_2})|$ with (θ_1, θ_2) at or very near $(\pm\pi, \pm\pi)$ would result in a division of zero by zero so the cost function would not be computable and the value ∞ could be assigned. In this example, however, if the evaluation of $|F|$ is not near enough to $(\pm\pi, \pm\pi)$ for the denominator to have a value of exactly zero in the computer, the numerator will have a value approximately the same as the denominator, and the computation would produce $|F| < \epsilon$ for some $\epsilon > 0$. In a CDC 6400, the denominator would have to have magnitude less than 10^{-293} before the value zero would be assigned. Therefore, unless the cost function requires calculation of $|F|$ extremely close to a singularity, nothing in the computation will interfere with the attempted convergence to an unstable filter as described in the example.

[2] The points $(z_1 = e^{\pm j\pi}, z_2 = e^{\pm j\pi})$ are nonessential singularities of the second kind. For a complete discussion of the singularities of functions of several complex variables, see [3].