and, for $dt > 0$,

$$\sum_{i=0}^{N} \frac{d\mu_i(t)}{dt} d\lambda_i(t) < 0.$$

Since $d\lambda_i(t) = (\lambda_i^2 - \lambda_i^1)dt$, it follows that

$$\sum_{i=0}^{N} \frac{d\mu_i(t)}{dt} (\lambda_i^2 - \lambda_i^1)dt = \sum_{i=0}^{N} (\lambda_i^2 - \lambda_i^1)d\mu_i(t) < 0.$$

Then, by integration over the defined range of t,

$$0 > \int_0^1 \sum_{i=0}^{N} (\lambda_i^2 - \lambda_i^1)\, d\mu_i(t) = \sum_{i=0}^{N} (\lambda_i^2 - \lambda_i^1)[\mu_i(1) - \mu_i(0)].$$

But, by assumption, $\mu_i(1) = \mu_i(0)$; and so

$$0 > \sum_{i=0}^{N} (\lambda_i^2 - \lambda_i^1)[\mu_i(1) - \mu_i(0)] = 0,$$

which is a contradiction. It therefore follows that the assumption is untenable, and the theorem has been proved.

IV. Numerical Method

The theorems of Section II suggest the following procedure for finding the maximum-entropy distribution for a given moment vector.

1) Solve first the one-moment problem (which can be accomplished directly since $(1,\mu_1) \Leftrightarrow (\lambda_0 = \ln \mu_1, \lambda_1 = 1/\mu_1)$).

2) After having solved the $(i-1)$-moment problem, test, by means of the criteria provided in Section II, whether the i-moment problem has a solution or not.

3) If the i-moment problem satisfies the criteria of Section II, define a "path" of discrete points between $(1,\mu_1,\cdots,\mu_{i-1},\mu_{i,\max})$ and $(1,\mu_1,\cdots,\mu_i)$ in the μ-space (for instance, the points $(1,\mu_1,\cdots,\mu_i + n(\mu_{i,\max} - \mu_i)/10)$, for $n = 0,\cdots,10$) and solve, by means of the first-order gradient method, the i-moment problem for the points on this path for successively increasing n. For each iteration, take the solution vector for the previous point as the starting point.

4) If the i-moment problem does not satisfy the criterion of Theorem 2, define a path similar to the one of step 3) between the image of $(\lambda_0,\cdots,\lambda_{i-1},0,0)$ and the point $(1,\mu_1,\cdots,\mu_{i-1})$, and proceed as in step 3. $(\lambda_0,\cdots,\lambda_{i-1})$ is the solution of the $(i-1)$-moment problem.

5) Continue until the original problem is solved (or until the procedure breaks down).

In short, then, the procedure is a combination of the first-order gradient method and the continuation method.

V. Some Numerical Results

The procedure of Section IV has been implemented on a CDC 6400 computer and has performed satisfactorily for problems of up to eight moments. For moment vectors of the chi-square distribution, for example, it could be demonstrated that for $m = 1.5$, where m is the degree of freedom, equations (1) and (2) have no solutions for $N = 2,4,6$; for $m = 3,4,5,6,7,8$, there were no solutions for $N = 3,5,7,9$. For all intermediate values of N, the procedure converged and produced solutions within specified tolerance margins. Table I shows the results of a run with $m = 4$ and $N = 2,4,6,8$, and Fig. 1 shows a plot of the fitted curves for these cases with, for comparison, plots of the original curve. The calculated entropies and the shape of the fitted curves are consistent with values tabulated by Wilson and Wragg in [4].

References

[1] A. Wragg and D. C. Dowson, "Fitting continuous probability density functions over $[0,\infty)$ using information theory ideas," *IEEE Trans. Inform. Theory*, vol. IT-16, pp. 226–230, Mar. 1970.

[2] D. C. Dowson and A. Wragg, "Maximum entropy distributions having prescribed first and second moments," *IEEE Trans. Inform. Theory*, vol. IT-19, pp. 689–693, Sept. 1973.

[3] J. P. Noonan, N. S. Tzannes, and T. Costello, "On the inverse problem of entropy maximizations," *IEEE Trans. Inform. Theory*, vol. IT-22, pp. 120–123, Jan. 1976.

[4] G. A. Wilson and A. Wragg, "Numerical methods for approximating continuous probability density functions, over $[0,\infty)$, using moments," *J. Inst. Maths. Applics.* 12, pp. 165–173, 1973.

There Is no MacWilliams Identity for Convolutional Codes

JAMES B. SHEARER AND ROBERT J. McELIECE, MEMBER, IEEE

***Abstract*—An example is provided of two convolutional codes that have the same transmission gain but whose dual codes do not. This shows that no analog of the MacWilliams identity for block codes can exist relating the transmission gains of a convolutional code and its dual.**

The *transmission gain* of a convolutional code is defined to be the power series $A(z) = \Sigma_{n=0}^{\infty} a_n z^n$, where a_n is the number of paths of Hamming weight n through the code's state diagram that begin and end in the zero state but do not otherwise pass through the zero state. (By convention we set $a_0 = 1$.) If a block code is viewed as a convolutional code with zero memory, the transmission gain is identical to the code's weight enumerator. Since the MacWilliams identity [2, chap. 16] shows that the weight enumerator of a block code is uniquely determined by the weight enumerator of its dual code, it is natural to wonder whether an analogous result holds for transmission gains for convolutional codes. The following simple example shows that it does not.

Let C_1 and C_2 be the (3,1) convolutional codes with generator matrices

$$G_1 = [1,D,1+D] \qquad G_2 = [D,D,1+D],$$

and let $C_1^\perp, C_2^\perp$ be the corresponding dual codes (see [3] for definitions). Then the dual codes $C_1^\perp$ and $C_2^\perp$ can be generated by

$$H_1 = \begin{bmatrix} 1 & 1 & 1 \\ D & 1 & 0 \end{bmatrix} \qquad H_2 = \begin{bmatrix} 1 & 1 & 0 \\ 1+D & 0 & D \end{bmatrix}.$$

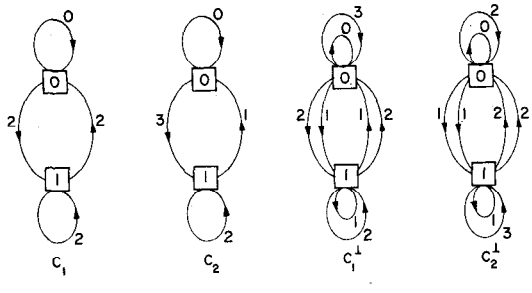

Fig. 1. State diagrams for $C_1, C_2, C_1^\perp, C_2^\perp$.

All of these matrices yield minimal encoders for their respective codes, and each encoder has only two states which we label 0 and 1. In Fig. 1, we have drawn the corresponding state diagrams in

Manuscript received November 5, 1976; revised March 4, 1977. This paper presents the results of one phase of research carried out at the Jet Propulsion Laboratory, California Institute of Technology, under Contract No. NAS 7-100, sponsored by the National Aeronautics and Space Administration.

J. B. Shearer was with the Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA. He is now with the Math Department, Massachusetts Institute of Technology, Cambridge, MA 02139.

R. J. McEliece is with the Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA 91103.

which each edge is labeled with the Hamming weight of the corresponding encoder output. Now, using standard combinatorial techniques (see, e.g., [1, sec. 5.6]), it is easy to verify that the transmission gains are

$$A_1(z) = A_2(z) = \frac{1 - z^2 + z^4}{1 - z^2}$$

$$A_1^{\perp} = \frac{1 - z + 3z^3 - z^5}{1 - z - z^2} = 1 + z^2 + 4z^3 + 5z^4 + \cdots$$

$$A_2^{\perp} = \frac{1 - z + z^2 + 2z^3 - z^5}{1 - z - z^3} = 1 + z^2 + 4z^3 + 4z^4 + \cdots.$$

REFERENCES

[1] A. Aho, J. Hopcraft, and J. Ullman, *The Design and Analysis of Computer Algorithms*. Reading, MA: Addison-Wesley, 1974.
[2] E. R. Berlekamp, *Algebraic Coding Theory*. New York: McGraw-Hill, 1968.
[3] G. D. Forney, "Convolutional codes I: Algebraic structure," *IEEE Trans. Inform. Theory*, vol. IT-16, pp. 720–738, November 1970.

High-Radix Transforms for Reed–Solomon Codes over Fermat Primes

KUANG Y. LIU, MEMBER, IEEE, IRVING S. REED, FELLOW, IEEE, AND T. K. TRUONG

***Abstract*—It is shown that a high-radix fast Fourier transform (FFT) with generator $\gamma = 3$ over GF(F_n), where $F_n = 2^{2^n} + 1$ is a Fermat prime, can be used for encoding and decoding of Reed–Solomon (RS) codes of length 2^{2^n}. Such an RS decoder is considerably faster than a decoder using the usual radix 2 FFT. This technique applies most ideally to a 16-error-correcting, 256-symbol RS code of 8 bits being considered currently for space communication applications. This special code can be encoded and decoded rapidly using a high-radix FFT algorithm over GF(F_3).**

I. INTRODUCTION

In this correspondence, a method is presented to streamline the transform decoding algorithm for Reed–Solomon (RS) codes, discussed in [1], [2]. This method uses transforms over GF(F_n), where $F_n = 2^{2^n} + 1$, for $n = 1,2,3,4$, is a Fermat prime [3], [4]. Basically, to decode short RS codes over GF(F_n), one can use $\gamma = \sqrt{2}$ as the generator of the transform. The maximum possible transform and code lengths for this generator are only 2^{n+2} [4]. However, the arithmetic used to perform these transforms is simple and requires only integer additions and circular shifts [5], [6].

To obtain longer RS codes on GF(F_n), one must use a generator γ with a higher multiplicative order. The generator with highest order is a primitive element. For example, it is well-known that $\gamma = 3$ is a primitive element in GF(F_n). If such a γ is used as the generator of the transform, the maximum transform length of 2^{2^n} can be achieved. Thus RS codes of as many as 2^{2^n} symbols of 2^n bits each can be generated on GF(F_n). Since the order of γ is a power of 2, a fast Fourier transform (FFT) type algorithm exists on GF(F_n) and can be used to decode such a 2^{2^n} symbol RS code.

The arithmetic used to perform transforms over GF(F_n), using $\gamma = 3$ as the generator, requires integer multiplications by powers of 3 and integer additions modulo F_n. But integer multiplications by powers of 3 modulo F_n are not as simple computationally as multiplications by powers of 2 modulo F_n. Since multiplication modulo F_n is the most time-consuming and complicated operation used, it is desirable to reduce the multiplications in order to increase the speed and/or to decrease the complexity of the RS decoder. Towards this end, for $\gamma = 3$, a high-radix FFT is developed to reduce the number of multiplications modulo F_n required to decode these longer RS codes on GF(F_n).

In space communication-link applications, a concatenated coding scheme was proposed [7] to reduce the value of the signal-to-noise ratio required to meet a specified bit-error rate. Such a coding scheme involves the concatenation of a standard 16-error-correcting, 256-symbol RS code with either a $K = 7$ rate $\frac{1}{2}$ or a $K = 7$ rate $\frac{1}{3}$, Viterbi decoded, convolutional code, where each symbol of the RS code is 8 bits. Such an RS code can be approximated by another RS code of 2^8 symbols of 8 bits each where each code symbol is an element of GF(F_3). The latter code can use the above-mentioned number-theoretic FFT on GF(F_3) to encode and decode the 2^8 symbols of the code.

The parity check symbols in this 256-symbol RS code may occur anywhere in the range between 0 and 2^{2^3}. If 2^{2^3} is observed as a parity check symbol, deliberately change this symbol to 0, now an error. The decoder will then automatically correct this error. Hence an RS code, which is generated in GF(F_3), can replace the original RS code used in concatenation with the convolutional code for this application.

It is shown in this note that an RS code generated in GF(F_3) can be decoded using either a radix 4 or a radix 16 FFT over GF(F_3). This requires, respectively, 30 or 70 percent fewer modulo F_n multiplications than the radix 2 FFT over GF(F_n) (see Table I).

II. HIGH-RADIX FFT ALGORITHM OVER GF(F_n) WHERE F_n IS A FERMAT PRIME

The transform over GF(F_n) is of the form

$$A(f) = \sum_{t=0}^{d-1} a(t)\gamma^{ft}, \qquad \text{for } 0 \le f \le d-1, \tag{1}$$

where $F_n = 2^{2^n} + 1$ is a Fermat prime for $n \le 4$. In (1), the transform length d divides $F_n - 1$, $a(t) \in$ GF(F_n), and γ is a primitive dth root of unity which generates the d element cyclic subgroup

$$G_d = \{\gamma, \gamma^2, \cdots, \gamma^{d-1}, 1\}$$

in the multiplicative group of GF(F_n). The inverse transform of

TABLE I
NUMBER OF OPERATIONS REQUIRED TO TRANSFORM $d = 256$ POINT FFT OVER GF(F_n), WHERE $n = 3,4$

Algorithm	Mod F_n Multiplications	Mod F_n Additions	Circular Shifts
Radix 2 ($d = 2^8$)	769	2048	0
Radix 4 ($d=(2^2)^4$)	513	2048	256
Radix 16 ($d=(2^4)^2$)	225	2048	544

Manuscript received December 13, 1976. This work was supported in part by NASA Contract Nos 7-100, and also in part by the United States Air Force Office of Scientific Research under Grant AFOSR-75-2798.

K. Y. Liu and I. S. Reed are with the Department of Electrical Engineering, University of Southern California, Los Angeles, CA 90007.

T. K. Truong was with the Department of Electrical Engineering, University of Southern California, Los Angeles, CA 90007. He is now with Jet Propulsion Laboratory, CA 91103.