FIR Parity Check Codes

Chris Heegard, *Fellow, IEEE,* and Andrew J. King

***Abstract*—This paper describes a method for packet synchronization and error detection for use in a synchronous digital communications system. The method relies upon a class of linear block codes that have parity checks that are expressed in terms of a *finite-impulse response* (FIR) filter. This system is incorporated in the newly established ITU standard of digital cable television standard, J.83 appendix B, which is based on an MPEG 2 transport packet data stream. This technique is also the basis for cable modem downstream transmission defined in the IEEE 802.14 and MCNS standards. The parity check structure is based on a pseudonoise sequence generated by a (binary) primitive polynomial. This structure allows for a computationally efficient implementation of the parity check FIR filter, in a recursive manner, that is none the less self-synchronizing. The FIR parity check codes that are described are characterized as the dual of a CRC-type, shortened cyclic code. The theory and computational structure of these codes are presented here; the J.83B code is used as an example of the general theory.**

***Index Terms*—Block codes, cyclic codes, dual codes, linear codes, pseudonoise coding, FIR digital filters.**

I. Introduction

THIS paper considers a class of linear block codes (LBC's) that have parity checks that are expressed in terms of a *finite-impulse response* (FIR) filter. The codes are useful for block synchronization and error detection in a synchronous digital communications system. The objective of the block encoding is to periodically insert parity check information into the packet stream. Initially, the parity checks are used to establish packet boundaries at the receiver in order to parse the data into the desired packets. After packet synchronization is established, the parity checks are used to detect uncorrected errors that may have occurred at the output of the receiver/error correction processing. This system is incorporated in the newly established ITU standard for digital cable television standard, J.83 appendix B, which is based on an MPEG 2 transport packet data stream [1]. This technique is also the basis for cable modem downstream transmission defined in the IEEE 802.14 and MCNS standards.

The novel aspect of the encoding is that the parity checks of the blockcode are computed at the receiver by observing the output of a FIR linear time-invariant, (binary) filter. The parity check structure is based on a pseudonoise (PN) sequence generated by a (binary) primitive polynomial. This structure allows for a computationally efficient implementation of the parity check FIR filter, in a recursive manner, that is none the less self-synchronizing. The FIR parity check codes are characterized as the dual of a CRC-type, shortened cyclic code.

The paper begins in Section II with background material that reviews basic definitions and ideas from algebraic coding theory and the theory of polynomials and filters. The second section, Section III, describes the theory and computational structure of FIR parity check codes. In Section IV, a few small examples, as well as the J.83B code are used to illustrate the general theory. The reader may find it helpful to peruse the examples in Section IV in conjunction with understanding the details presented in Section III.

II. Background

A. LBC's

A binary LBC $\mathcal{C} \subset \mathcal{F}_2^n$ with parameters (n, k) consists of a k dimensional subspace of the set of binary n-tuples $\mathcal{F}_2^n$ [2]–[4]. Such a space can be described by a set of k *basis vectors* placed in a $k \times n$ *generator matrix*

$$G = \begin{pmatrix} g_{1,1} & g_{1,2} & \cdots & g_{1,n} \\ g_{2,1} & g_{2,2} & \cdots & g_{2,n} \\ \vdots & \vdots & \cdots & \vdots \\ g_{k,1} & g_{k,2} & \cdots & g_{k,n} \end{pmatrix}.$$

Given a generator matrix, a (linear) encoder can be constructed via matrix multiplication $\mathbf{c} = \mathbf{m}G$, where $\mathbf{m} \in \mathcal{F}_2^k$ is a binary k-tuple. If the first k columns of G is the identity matrix, we say that G represents a *systematic encoder*. Note that a given LBC $\mathcal{C}$ has many generator matrices and thus, many encoders.

An alternate description of a linear code $\mathcal{C}$ is given by a set of $r = n - k$ linearly independent *parity-check equations* that can be expressed in terms of a *parity-check matrix* H, an $r \times n$ binary matrix. A binary vector of length n is a *codeword* if and only if

$$(c_0, c_1, \cdots, c_{n-1}) \begin{pmatrix} h_{1,1} & h_{2,1} & \cdots & h_{r,1} \\ h_{1,2} & h_{2,2} & \cdots & h_{r,2} \\ \vdots & \vdots & \cdots & \vdots \\ h_{1,n} & h_{2,n} & \cdots & h_{r,n} \end{pmatrix} = (0, 0, \cdots, 0)$$

or $\mathbf{c}H^t = \mathbf{0}$. Again, a given LBC $\mathcal{C}$ has many parity-check matrices.

The space spanned by the rows of a given parity-check matrix H for a code $\mathcal{C}$ form an (n, r) LBC called the *dual code*. This code is described by

$$\mathcal{C}^{\perp} = \{\mathbf{d} \in \mathcal{F}_2^n | \mathbf{d} \cdot \mathbf{c} = 0 \text{ for all } \mathbf{c} \in \mathcal{C}\}.$$

Paper approved by J. Huber, the Editor for Coding and Coded Modulation of the IEEE Communications Society. Manuscript received September 23, 1997; revised October 15, 1997. The work of C. Heegard was supported in part by the National Science Foundation under Grant NCR-9207331 and Grant NCR-9520981.

C. Heegard is with Alantro Communications, Santa Rosa, CA 95401 USA (e-mail: heegard@alantro.com).

A. J. King is with Advanced Micro Devices, Austin, TX 78610 USA.

Publisher Item Identifier S 0090-6778(00)06152-3.

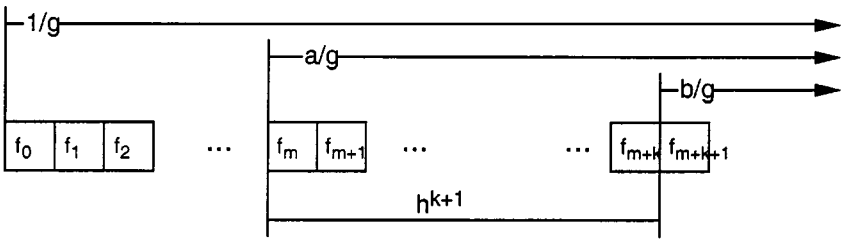

Fig. 1. The relationship between $f(x) = 1/g(x)$, $a(x)/g(x)$ and $b(x)/g(x)$.

A *coset* of a LBC is obtained via solutions to the equation

$$\mathbf{c}H^t = \mathbf{S}$$

where $\mathbf{S}$ is a fixed binary vector of length r.

To encode onto a coset of a LBC, a constant vector can be added

$$\mathbf{c} = \mathbf{m}G + \sigma$$

where $\sigma H^t = \mathbf{S}$. For a systematic encoder, the constant vector σ can always be chosen to be zero in the first k coordinates (thus a constant is added to the parity checks only).

B. Polynomials, Rational Functions, and Filters

Given a binary polynomial

$$g(x) = 1 + g_1 x + g_2 x^2 + \cdots + g_{r-1}x^{r-1} + x^r$$

of degree r, the power series

$$\frac{1}{g(x)} = f(x) = \sum_{j=0}^{\infty} f_j x^j$$

is defined by the relationship

$$g(x)f(x) = 1.$$

Note that the power series $f(x)$ forms a periodic sequence, that is

$$f(x) = f^P(x) \sum_{i=0}^{\infty} x^{iP} = \frac{f^P(x)}{1 - x^P}$$

where P is the (fundamental) period of $f(x)$ and $f^P(x) = \sum_{j=0}^{P-1} f_j x^j$ represent the truncated power series with P terms. It is well known that the period $P \leq 2^r - 1$, where equality is achieved if and only if $g(x)$ is a *primitive* polynomial; in this case, $f(x)$ is said to be a *maximal length, pseudorandom* (PN) sequence.

Closely associated with the power series $f(x)$ is the *difference* equation

$$y_j = w_j - \sum_{l=1}^{r} g_l y_{j-l} = \sum_{l=0}^{\infty} f_l w_{j-l}$$

which describes a *linear, time-invariant, causal, finite-state* system with input sequence w_j, output sequence y_j, and impulse response f_j.

Similarly, a rational function (i.e., ratio of polynomials)

$$\frac{a(x)}{g(x)} = e(x) = \sum_{j=0}^{\infty} e_j x^j$$

is defined by the polynomial equation

$$g(x)e(x) = a(x).$$

The power series $e(x)$ implies the difference equation

$$y_j = \sum_{m=0}^{\text{degree}(a)} a_m w_{j-m} - \sum_{l=1}^{r} g_l y_{j-l} = \sum_{l=0}^{\infty} e_l w_{j-l}.$$

If $f(x) = 1/g(x)$ and $m \geq 0$ is any nonnegative integer then the *delayed* power series $(e_j = f_{j+m})$

$$e(x) = \sum_{j=0}^{\infty} f_{j+m} x^j = \frac{a(x)}{g(x)}$$

for some $a(x)$ of degree less than r ($a(x)$ is easily determined from the polynomial equation $g(x)e(x) = a(x)$).

Furthermore, for the nonnegative integer $n \geq 0$ the *truncated* power series (a polynomial) $(e_j = 0$, for $j > n)$

$$h^{k+1}(x) = \sum_{j=0}^{k} f_{j+m} x^j = \frac{a(x) - x^{k+1}b(x)}{g(x)}$$

for some $b(x)$ of degree less than r. In fact

$$\frac{b(x)}{g(x)} = \sum_{j=0}^{\infty} f_{j+m+k+1} x^j$$

$b(x)$ is easily determined from the polynomial equation $g(x)h^{k+1}(x) = a(x) - x^{k+1}b(x)$. The difference equation, in this case

$$y_j = \sum_{m=0}^{\text{degree}(a)} a_m w_{j-m} - \sum_{t=0}^{\text{degree}(b)} b_t w_{j-(k+1)-t} \sum_{l=1}^{r} g_l y_{j-l}$$
$$= \sum_{l=0}^{k} h_l w_{j-l}, \tag{1}$$

which corresponds to a *sliding window* or FIR system.

The relationship of three power series is indicated in Fig. 1.

We are interested in the polynomial $h^{k+1}(x)$, of degree k, that is obtained from the subsequence of the power series $f(x)$ starting with the term f_m up to the term f_{m+k}. Since $a(x)/g(x)$ begins with f_m, and $b(x)/g(x)$ begins with f_{m+k+1}, the piece we are interested in can be obtained by shifting $b(x)/g(x)$ by

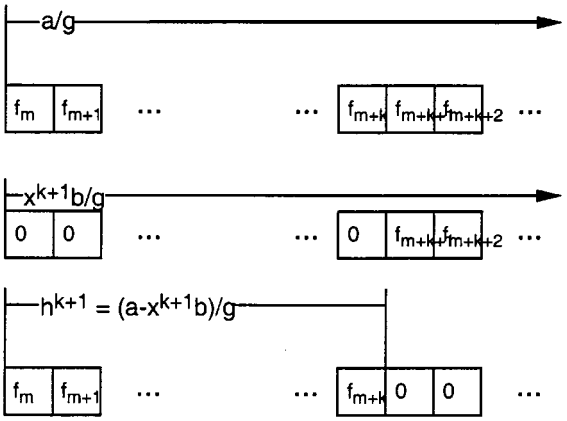

Fig. 2. The relationship between $a(x)/g(x)$, $x^{k+1}b(x)/g(x)$ and $h^{k+1}(x)$.

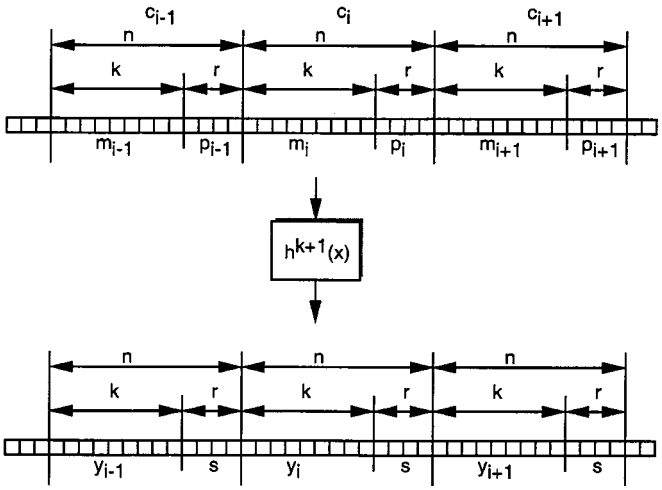

Fig. 3. Syndrome calculation.

$k+1$ places (i.e., by multiplying by x_{k+1}) and subtracting. This relationship is depicted in Fig. 2. Note that since $a(x)$ has a degree less than $r(< k)$, the first r terms of the sequence, $f_m, f_{m+1}, \cdots, f_{m+r-1}$, form $a(x)$, while the r terms starting at x^{k+1} form $b(x)$ (there is no overlap in the difference $a(x) - x^{k+1}b(x)$). Note that this filter can be implemented, using the difference equation (1), with a number of terms that has $\|a(x)\| + \|b(x)\| + \|g(x)\|$ terms rather than $\|h^{k+1}(x)\|$ terms (where $\|\cdot\|$ is the *Hamming weight* of the polynomial). This is very important in our application, since the polynomials $a(x)$, $b(x)$, $g(x)$ are small degree (8 or less) and the degree of $h^{k+1}(x)$ is large.

III. FIR Parity-Check LBC's

We are interested in generating LBC's for which the parity check equations can be implemented by a single FIR filter. Let $h^{k+1}(x)$ be a polynomial of degree k with nonzero constant term $(h_0 = h_k = 1)$. Then, the equation shown at the bottom of the page defines an {FIR-parity-check-code} FIR-PCC. With such an FIR-parity-check LBC, codewords $c_i(x)$ that are concatenated together

$$w(x) = \sum_{i=0}^{\infty} c_i(x)x^{in}$$

can be synchronized by passing $w(x)$ through the FIR filter with response $h^{k+1}(x)$

$$S(x) = h^{k+1}(x)w(x) = \sum_{i=0}^{\infty} (y_i(x) + x^k s(x))x^{in}$$

where each $y_i(x)$ has a degree less than k (see Fig. 3). The *syndrome sequence* $S(x)$ has the constant polynomial $s(x)$ (a polynomial of degree less than r that determines which coset of the LBC is used) periodically embedded at the end of each n-block (in the absence of channel errors). Thus by correlating the syndrome sequence $S(x)$ with the constant polynomial $s(x)$, synchronization can be established and maintained. It is important in the acquisition phase that the correlation with the known $s(x)$ be reliable in the presence of noise (channel errors). For this reason, care should be taken in the design of "sync" polynomial $s(x)$ to address this issue. In particular, the polynomial should have good autocorrelation properties (e.g., a Barker code). In fact, the zero polynomial $s(x) = 0$ (which corresponds to a linear code) has a poor autocorrelation and thus would be a poor choice.

By using a systematic encoder, the data is recovered directly from the concatenated sequence $w(x)$. Furthermore, once synchronization has been established, the syndrome can be used to detect errors in the data whenever the fixed polynomial $s(x)$ fails to appear at the anticipated time. Finally, by using an FIR parity check means that the syndrome former is self-synchronizing (i.e., the initial conditions of the syndrome are resolved automatically) and has limited error propagation (the effect of an error is limited to the length of the FIR window, $k+1$ bits).

To choose the response $h^{k+1}(x)$, a recursive solution is required. Once a recursion polynomial $g(x)$ is selected, a suitable $a(x)$ and $b(x)$ are determined so that

$$h^{k+1}(x) = \frac{a(x) - x^{k+1}b(x)}{g(x)}$$

with nonzero constant term and degree k $(h_0 = h_k = 1)$. (Note that once $g(x)$ is specified, only certain choices of $a(x)$ and

$$H = \begin{pmatrix} h_k & h_{k-1} & \cdots & \cdots & \cdots & h_0 & 0 & 0 & \cdots & 0 \\ 0 & h_k & \ddots & \cdots & \cdots & h_1 & h_0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \cdots & \vdots & \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & \cdots & h_k & \cdots & h_{r-1} & h_{r-2} & h_{r-3} & \cdots & h_0 \end{pmatrix}$$

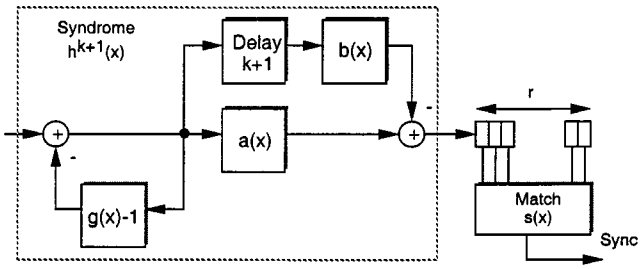

Fig. 4. Syndrome calculator/sync detect.

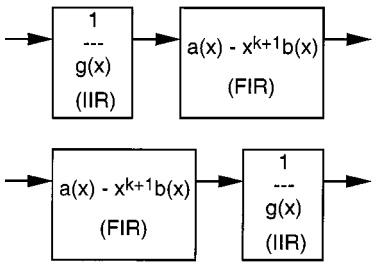

Fig. 5. Two ways to cascade the IIR and FIR filters.

$b(x)$ will work; namely those for which $g(x)$ divides $a(x) - x^{k+1}b(x)$.)

Fig. 4 shows the general form for the calculation of the syndrome and synchronization detection. It is important to note that the effects of the initial conditions in the recursive portion of the circuit (i.e., the feedback part that implements the $1/g(x)$ function) have finite effect on the output. This is a direct consequence of the fact that the polynomial $f(x) = a(x) - x^{k+1}b(x)$ is divisible by $g(x)$ and that the operations that implement these functions operate on the output of the recursive part of the circuit. For example, if the initial conditions are nonzero in the $1/g(x)$ circuit, and the input is constantly zero, then the output of the recursive portion is the PN sequence generated by $1/g(x)$. However, within n steps, the syndrome calculator will constantly produce zero at the output. This property also insures that the effects of channel errors on the output of the syndrome will be restricted to the $k+1$ span of the impulse response of the syndrome function.

The output of the syndrome filter is shifted into a register of length r where a match to the desired sync pattern $s(x)$ is made. This sync signal is then used to establish and maintain synchronization as well as to detect the presence of errors.

One of the more interesting aspects of this approach relates to the structure of the syndrome checker. First, we note that the decoder acts as a filter that is continually fed inputs and produces outputs. It needs never to be reset. It is self-synchronizing and naturally limits error propagation (these two concerns are of course intimately related). For simplicity of understanding, consider the startup problem with an offset of 0. Assume the data is always zero (thus the parity checks are all 0) and this is presented at the input of the decoder. Furthermore, assume that has been randomly initialized with a nonzero value. Since the input of the decoder is zero, we would like the output of the decoder to produce the all 0's output after a short transient due to the initial conditions of the decoder circuit. This is in fact what will happen (even though the state of the recursive part of the circuit will never go to the zero state, in this case). Why is this? A nonzero initial condition in the denominator circuit, with an all 0's input, will in fact produce the PN sequence associated with $g(x)$. That is, the output will be the power series $f(x)$ with some random delay M (which depends on the exact choice of initial condition). Thus, we know that the output sequence can be written as a rational function $c(x)/g(x)$, where the degree of $c(x)$ is less than r (the degree of $g(x)$). However, after the numerator circuit is applied, the output becomes a polynomial of degree at most $(r-1)+(k+1) = n$ (the blocklength of the code). (Algebraically, this follows from the fact that $g(x)$ divides $a(x) - x^{k+1}b(x)$.) This means, that within one blocklength, the output of the decoder parity check circuit will become all 0s (even though the state of the parity check circuit is nonzero forever). This argument can be extended to nonzero data, nonzero offset and to account for the finite error propagation in a straightforward manner.

It is important to note that from an impulse response (or input/output) analysis, the two cascaded filters in Fig. 5 are the same. The two ways implement the same rational transfer function, but from a transient (or initial condition) analysis, the two act quite differently. The first form is self-synchronizing, the latter is not.

Finally, a systematic encoder must be implemented. A simple two pass encoder is constructed as follows. First, take the k bit message, followed by r zeros, and pass the resulting n bits through a syndrome calculator, producing an n bit output. Throw out the first k outputs and retain the last r bits. These r bits are then passed through a filter with response $g(x)/a(x)$ (this method assumes $k > r$). The r bits produced from this last filtering, summed with a constant r bits (that determines $s(x)$), determine the parity bits that are to be transmitted. Note that there are simplifications that can be applied here. Firstly, only the first $r-1$ message bits are needed for the "$b(x)$"; there is no need to build a buffer of length $k+1$ (as required at the receiver).

A. Relationships with Cyclic Codes and CRCs

A common class of (n, k) LBC's used for error detection in a large variety of transmission and storage systems are the *cyclic redundancy check* (CRC) codes. These codes are best described in terms of polynomial codewords $c(x) \in \mathcal{F}_2^n[x]$ (the set of binary polynomials of degree $< n$) and a *generator polynomial* $g(x)$ of degree $r = n - k$. The code

$$\mathcal{C} = \{c(x) \in \mathcal{F}_2^n[x] | c(x) = m(x)g(x), m(x) \in \mathcal{F}_2^k[x]\}$$

is all polynomial multiples of the generator polynomial $g(x)$ of degree less than n. A CRC code can be described as an intersection of the *ideal* generated by $g(x)$ in the *ring* of polynomials $\mathcal{F}_2[x]$ [5]

$$\begin{aligned}\mathcal{C} &= \langle g(x)\rangle \cap \mathcal{F}_2^n[x] \\ \langle g(x)\rangle &= \{c(x) \in \mathcal{F}_2[x] | c(x) \\ &\quad = m(x)g(x), m(x) \in \mathcal{F}_2[x]\}.\end{aligned}$$

Given the generator $g(x)$, of degree r, and the blocklength n for a CRC, one can always find polynomials $h(x)$, of degree

TABLE I
A SMALL EXAMPLE, $g(x) = 1 + x + x^3$, $n = 6, 7, 8$

n	k	$a(x)$	$b(x)$	$f(x)$	$h(x)$	m
6	3	$1+x$	$1+x^2$	$1+x+x^4+x^6$	$1+x^3$	7
7	4	1	x^2	$1+x^7$	$1+x+x^2+x^4$	7
8	5	$1+x$	x^2	$1+x^3+x^8$	$1+x^3$	14

$k = n - r$ and $f(x)$ of degree n such that $g(x)h(x) = f(x)$ and then

$$\mathcal{C} = \{c(x) \in \mathcal{F}_2^n[x] | c(x)h(x) = 0 \text{ modulo } f(x)\}$$

and $\mathcal{C}$ forms an ideal in the *quotient ring* $\mathcal{F}_2[x]/\langle f(x)\rangle$. If one can solve $g(x)h(x) = x^n - 1$ (i.e., $f(x) = x^n - 1$) then the CRC is a *cyclic code*. In this case, the dual code is also a cyclic code with generator $x^k h(x^{-1})$

$$\mathcal{C}^{\perp} = \langle x^k h(x^{-1})\rangle \cap \mathcal{F}_2^n[x]$$

and thus the dual of a cyclic code is a cyclic code. However, for most choices of blocklength n, it is not possible to solve $g(x)h(x) = f(x)$ with $f(x) = x^n - 1$. For example, if $g(x)$ is a primitive polynomial (or the product of a primitive polynomial with $x - 1$, a common form of the generator used in many CRC codes), then the blocklength n must be divisible by $2^r - 1$ $(2^{r-1} - 1)$. For such other values of n, the code $\mathcal{C}$ is in fact a *shortened cyclic code* [2], [3] since one can always find an integer $m \geq n$ such that $g(x)$ divides $x^m - 1$ and the code $\mathcal{C}$ is obtained by shortening the $(m, m-r)$ code $\langle g(x)\rangle \cap \mathcal{F}_2^m[x]$ to the $(n, n-r)$ code $\mathcal{C} = \langle g(x)\rangle \cap \mathcal{F}_2^n[x]$.

For a shortened cyclic code, $m > n$, the polynomial $x^k h(x^{-1})$ does not generate the dual code and in fact the dual code is not the intersection with any ideal in $\mathcal{F}_2[x]$. Thus an FIR-PCC, which is the dual of a shortened cyclic code, is not a shortened cyclic code.

IV. APPLICATIONS

In the present application, it has been realized that for reasons of simplifying the syndrome calculation, the dual of a shortened cyclic code offers distinct advantages over the usual CRC technique of using a shortened cyclic code. Thus for example, once $g(x)$ and n have been fixed, the codewords of a CRC would be the set $\mathcal{C}_{\text{CRC}} = \langle g(x)\rangle \cap \mathcal{F}_2^n[x]$, while we propose to use the set $\mathcal{C}_{\text{FIR}} = (\langle x^k h(x^{-1})\rangle \cap \mathcal{F}_2^n[x])^{\perp}$. Note that $\mathcal{C}_{\text{FIR}} \neq \mathcal{C}_{\text{CRC}}$, unless the code is a cyclic code (i.e., $g(x)$ divides $x^n - 1$). The choice of $\mathcal{C}_{\text{FIR}}$ means that the simple syndrome calculation procedure described previously will apply; this would not be possible in general with $\mathcal{C}_{\text{CRC}}$.

A. Small Examples

As an example, consider the primitive polynomial $g(x) = 1 + x + x^3$ of degree $r = 3$ and the blocklengths $n = 6, 7, 8$. In the Table I, the dimension of the code k is given as well as $f(x) = a(x) - x^{k+1}b(x) = g(x)h(x)$ and the value of the smallest cyclic code length m.

Notice that only in the case $n = 7$ is the code a cyclic code, which has a dual that is itself a cyclic code. In the other two cases, $n = 6, 8$, the codes are shortened cyclic codes, that have duals that are not shortened cyclic codes.

B. The J.83 (appendix B)/ IEEE 802.14/MCNS, MPEG/Sync System

An FIR-PCC is incorporated in the newly established ITU standard of digital cable television standard, J.83 appendix B, which is based on an MPEG 2 transport packet data stream [1], [6]. This technique is also the basis for cable modem downstream transmission defined in the IEEE 802.14 and MCNS standards.

MPEG packets are 188 bytes ($n = 1504$ bits) long with 187 bytes ($k = 1496$ bits) of data and 1 byte ($r = 8$ bits) of sync-word. (Note that $2^8 - 1 = 255$ does not divide $n = 1504$.) The J.83B system [1] uses the space for the 1-byte sync-word to accomplished both synchronization and additional error detection (above that provided by the Reed–Solomon code in the FEC). This is accomplish via an FIR-parity-check based LBC (as described above). The parameters of the code are $(n = 1504, k = 1496)$ where

$$\begin{aligned} g(x) &= 1 + x + x^5 + x^6 + x^8 \\ a(x) &= 1 \\ b(x) &= 1 + x + x^3 + x^7 \end{aligned}$$

which is based on a primitive polynomial $g(x)$ of degree 8 (i.e., it produces a PN sequence of length 255), a constant $a(x)$, and a $b(x)$ with four terms.

The system uses a coset of the FIR-parity-check LBC. The standard uses the sequence $s(x) = 1 + x + x^2 + x^6$ (0x47 in Hex) that has good autocorrelation properties (in fact, it is the original MPEG sync-word!). This is obtained by adding the offset $\sigma(x) = 1 + x + x^2 + x^5 + x^6$ (0x67 in Hex) to the parity check byte at the transmitter.

V. CONCLUSIONS

This paper describes a new class of algebraic block codes that are described in terms of a syndrome that is easy to compute. The syndrome is calculated from the output of a FIR filter (over the binary field $\mathcal{F}_2$). Furthermore, the blocklength of the code can be quite long as the particular FIR structure described has, in fact, an IIR (or recursive) implementation which has little computational cost as a function of the blocklength.

The purpose of the FIR-PCC's are for synchronization and error detection. During an acquisition stage, the FIR-PCC is used to establish block boundaries in a fixed length packet (or frame) based system. Once the packet boundaries are established, the FIR-PCC code can be used to monitor for errors in transmission.

A key observation that makes the technique robust in the presence of noise is that the order of operation of a cascaded pair of filters, while irrelevant from an impulse response viewpoint, is critically important from a transient analysis perspective. Thus, with the correct configuration, the IIR implementation of the syndrome calculation has only finite error propagation at the output.

The application of the theory has been realized, in practice, in the international standard for digital television transmission (J.83b) that forms an integral part of the digital TV set-top box that is becoming wide-spread in the US and around the world. This same standard is used in the "downstream" transmission for the largest base of standards based cable modems.

ACKNOWLEDGMENT

The authors sincerely thank the reviewers for the detailed suggestions that have significantly improved the presentation of this material.

REFERENCES

[1] M. T. Irmer, "Digital multi-programme systems for television sound and data services for cable distribution,", ITU-T Recommendation J.83 ITU-T COM09 R R006E1.WW2, Int. Telecommun. Union, Nov. 2, 1995.
[2] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes*. Amsterdam, The Netherlands: North-Holland, 1977.
[3] S. Lin, J. Daniel, and J. Costello, *Error Control Coding: Fundamentals and Applications*. Englewood Cliffs, NJ: Prentice-Hall, 1983.
[4] S. B. Wicker, *Error Control Systems for Digital Communications and Storage*. Englewood Cliffs, NJ: Prentice-Hall, 1995.
[5] I. N. Herstein, *Topics in Algebra*, 2nd ed. New York: Wiley, 1975.
[6] C. Heegard, A. J. King, S. Lovely, and T. J. Kolze, "Synchronization and error detection in a packetized data stream," U.S. Patent 5 703 887, Dec. 30, 1997.

Chris Heegard was born in Pasadena, CA, on October 4, 1953. He received the B.S. and the M.S. degrees in electrical and computer engineering from the University of Massachusetts, Amherst, in 1975 and 1976, respectively, the Ph.D. degree in electrical engineering from Stanford University, Palo Alto, CA, in 1981.

From 1976 to 1978, he was a Research and Development Engineer at Linkabit Corporation, San Diego, CA, where he worked on the development of a packet switched satellite modem and several sequential decoders for the decoding of convolutional codes. In 1981, he joined the faculty of the School of Electrical Engineering, Cornell University, Ithaca, NY, as an Assistant Professor; he was appointed to Associate Professor, with tenure, and is currently a Full Professor. At Cornell, he teaches courses in digital communications, error control codes, information theory, detection and estimation theory, digital systems, and audio engineering. Since 1997, Dr. Heegard has been CEO and Principal Scientist of Alantro Communications, Inc., Santa Rosa, CA (www.alantro.com). Alantro Communications is a physical-layer communications hardware company with a particular expertise in forward-error correction (FEC). The technology is vital in high-speed wireless networking, cable modem/digital-tv, digital subscriber line, and satellite/terrestrial wireless. He is an active member of the consulting community. He has worked on problems of digital HDTV and cable TV transmission, DSP and hardware-based trellis coded modems, modulation and error-control for optical LAN's, and modulation and coding for recording systems. He is the inventor on several patents. He is the Founder and Chief Scientist for Native Intelligence (www.nativei.com), a digital communications company. His current research interests include information, coding and communication theory, algorithms for digital communications, coding for computer memory systems, signal processing and error-control in optical and magnetic recording systems, audio and video signal compression, algebraic geometric coding theory and symbolic and numerical computer methods.

In 1984, Dr. Heegard received the Presidential Young Investigator Award from the National Science Foundation and the IBM Faculty Development Award. He has ongoing research support from the NSF as well as ARO and NSA. He has been involved in the organization of several IEEE workshops and symposia. In 1986, he was elected to the Board of Governors of the Information Theory Society of the IEEE and reelected in 1989. In 1994, he was the President of the IT Society. He is a Fellow of IEEE, and a Member of AES and Eta Kappa Nu.

Andrew J. King is currently a Department Manager in the Communication Products Division of Advanced Micro Devices, developing systems technology for ADSL central office products. Previously, he has held positions of: Senior Staff Engineer,VLSI Engineering at Qualcomm, Inc., developing integrated circuits for CDMA wireless handsets; Design Group Manager, General Instrument Corporation Integrated Systems Center, developing digital transmission technology for cable systems; Sandia National Laboratories working on modeling and development of CMOS device and memory architectures; Hughes Aircraft Carlsbad Research Center working on high performance Bi-CMOS technology development; and SGS-Thomson Microelectronics, developing mixed signal and smart power devices in the computer peripherals IC design group.