Códigos de Bloco Lineares Baseados em Banco de Filtros e Wavelets Cíclicos sobre Corpos Finitos

G. J. da Silva Jr., R. M. Campello de Souza e H. M. de Oliveira

***Resumo*— Este artigo apresenta uma nova abordagem sobre códigos de bloco lineares utilizando banco de filtros e wavelets cíclicos. Estruturas em árvore para codificação e decodificação, baseadas em banco de filtros cíclicos de síntese e análise, são apresentadas. Em alguns casos particulares, é mostrado como se constroem as matrizes geradora e de paridade dos códigos. Códigos conhecidos são gerados por meio dessas estruturas, tais como os códigos de Hamming, Reed-Solomon e códigos de repetição. Novos tipos de códigos são encontrados através de um método de projeto baseado na transformada de Fourier de curta duração sobre corpos finitos.**

***Palavras-Chave*— Códigos de bloco lineares, códigos cíclicos, corpo finito, banco de filtros cíclicos, wavelets cíclicas, transformada Z cíclica, TFCF, TFCDCF.**

***Abstract*— This paper presents a new approach to linear block codes using cyclic filter banks and wavelets. Tree structures for encoding and decoding, based on synthesis and analysis cyclic filter banks, are presented. In some particular cases, the construction of the generator and parity-check matrices of the codes is shown. Known codes are generated from these structures, such as Hamming, Reed-Solomon and repetition codes. New types of codes are found through the design technique based on the finite field short time Fourier transform.**

***Keywords*— Linear block codes, cyclic codes, finite field, cyclic filter banks, cyclic wavelets, cyclic Z transform, FFFT, FFSTFT.**

I. INTRODUÇÃO

Ferramentas como a transformada discreta de Fourier (TDF), a transformada Z, banco de filtros e wavelets, são de grande importância para processamento de sinais sobre o corpo dos reais [1]-[3]. Estruturas sobre corpos finitos [4] vêm se tornando atrativas em telecomunicações para representação numérica, evitando problemas que ocorrem na representação de números reais em implementações digitais. A utilização de ferramentas de processamento de sinais sobre corpos finitos teve um grande impulso quando Pollard introduziu a transformada de Fourier de corpo finito (TFCF) [5]. Desde então, muitas ferramentas de engenharia vêm emergindo para estruturas definidas sobre corpos finitos [6]-[10], onde grande parte encontra aplicações em codificação de canal e em criptografia [11]-[14].

Banco de filtros cíclicos (BFC) e wavelets cíclicas [6] podem ser utilizados para análise, geração, codificação e decodificação de códigos de bloco. Cada filtro cíclico constitui per si um codificador cíclico, em função da utilização da convolução cíclica. Para o contexto de códigos, pode ser introduzida a denominação *banco de códigos*.

G. J. da Silva Jr., R. M. Campello de Souza e H. M. de Oliveira¸ Grupo de Processamento de Sinais, Departamento de Eletrônica e Sistemas, Universidade Federal de Pernambuco, Recife, PE, E-mails: gilsonjr@gmail.com, ricardo@ufpe.br, hmo@ufpe.br.

A notação indicada a seguir é utilizada neste trabalho:

$N \rightarrow$ Comprimento do código;
$K \rightarrow$ Dimensão do código (Comprimento da mensagem);
$d \rightarrow$ distância mínima do código;
C(N,K,d) $\rightarrow$ Representação dos parâmetros do código C;
$n \rightarrow$ Variável no domínio do tempo;
$k \rightarrow$ Variável no domínio da freqüência;
$c[n] \rightarrow$ Palavra código;
$u_i^{(j)}[n] \rightarrow$ Mensagem no canal i, estágio j;
$\tilde{c}[n] \rightarrow$ Palavra recebida do canal;
$\tilde{u}_i^{(j)}[n] \rightarrow$ Mensagem decodificada no canal i, estágio j;
$g_i[n] \rightarrow$ Filtro cíclicos de síntese do canal i;
$G_i(z) \rightarrow$ Transformada Z cíclica de $g_i[n]$;
$h_i[n] \rightarrow$ Filtro cíclicos de análise do canal i;
$H_i(z) \rightarrow$ Transformada Z cíclica de $h_i[n]$;
$GF(q) \rightarrow$ Campo de Galois de ordem q.

Transformadas sobre corpos finitos (a TFCF e a transformada Z cíclica) são utilizadas para a análise dos códigos e as estruturas BFC implementam sua codificação e decodificação. Todo deslocamento de seqüência representado neste artigo se refere ao deslocamento cíclico. Todos os polinômios no domínio da transformada Z cíclica estão sobre a aritmética módulo $(z^{-N} - 1)$.

Em um artigo recente [11] uma outra abordagem para a construção de códigos de bloco lineares por meio de banco de filtros foi apresentada, porém utilizando BFC diferentes daqueles apresentados em [10].

Este trabalho esta organizado em quatro seções. A seção II apresenta estruturas BFC para códigos de bloco lineares. Na seção III é proposto um método de projeto de códigos de bloco lineares baseado na transformada de Fourier de curta duração sobre corpos finitos. As conclusões do trabalho são apresentadas na seção IV e o Apêndice I descreve, de forma breve, alguns resultados importantes sobre sistemas cíclicos.

II. ESTRUTURAS BFC PARA CÓDIGOS DE BLOCO LINEARES

A idéia básica para a construção de códigos lineares é mostrada na figura 1. O código é gerado colocando mensagens $u_i^{(1)}[n]$ nas entradas dos filtros de síntese escolhidos e adicionando o sinal nulo para os outros filtros, gerando o sinal palavra código $c[n]$. Para decodificar $c[n]$, basta aplicá-lo ao banco de análise. As posições escolhidas como nulas estão

associadas às equações de paridade do código ($s_i^{(1)}[n] = 0$) e, na recepção, são denominadas de *síndrome*.

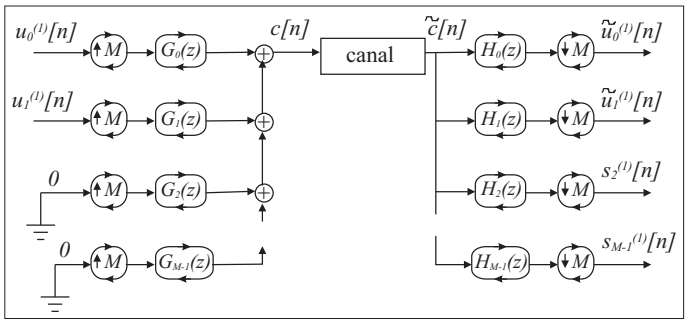

Fig. 1. Estrutura básica BFC com M canais.

Para simplificar a representação da estrutura básica, é utilizada a representação reduzida (em forma de árvores) de síntese e análise, mostradas nas figuras 2 e 3 respectivamente.

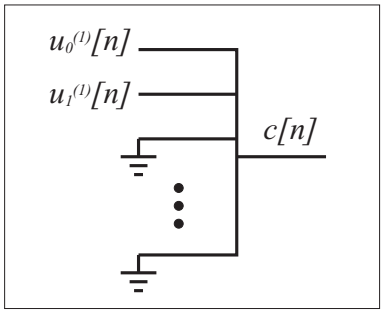

Fig. 2. Representação reduzida do banco de síntese

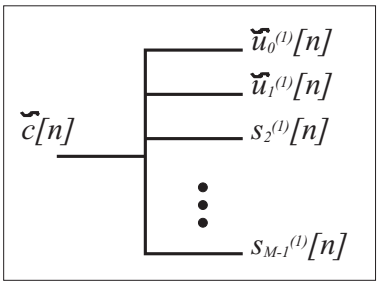

Fig. 3. Representação reduzida do banco de análise

O código de saída da estrutura pode ser analisado pela transformada Z cíclica, resultando em

$$C(z) = M^{-1} \sum_{i=0}^{M-1} U_i^{(1)}(z^M) G_i(z). \quad (1)$$

Aplicando a transformada inversa, tem-se

$$c[n] = \sum_{i=0}^{M-1} \sum_{l=0}^{N/M-1} u_i^{(1)}[l] g_i[n - Ml]. \quad (2)$$

Para a codificação, escolhe-se de forma geral um espaço código denotado por $\mathbf{C}$, que contém os índices i das mensagens $u_i^{(1)}[n]$ permitidas no código. Assim, uma palavra código $c[n] \in \mathbf{C}$ será expressa por

$$c[n] = \sum_{i \in \mathbf{C}} \sum_{l=0}^{N/M-1} u_i^{(1)}[l] g_i[n - Ml]. \quad (3)$$

Após $c[n]$ passar por um canal, recebe-se $\tilde{c}[n]$. Aplicando-se $\tilde{c}[n]$ ao banco de análise, cuja saída é dada por

$$u_i^{(1)}[l] = \sum_{n=0}^{N-1} c[n] h_i[Ml - n], \quad (4)$$

obtém-se as equações de decodificação (5) e síndrome (6),

$$\tilde{u}_i^{(1)}[l] = \sum_{n=0}^{N-1} \tilde{c}[n] h_i[Ml - n], \quad (5)$$

para $i \in \mathbf{C}$, e

$$s_i[l] = \sum_{n=0}^{N-1} \tilde{c}[n] h_i[Ml - n], \quad (6)$$

para $i \notin \mathbf{C}$. Se $\tilde{c}[n]$ pertence ao código, então $s_i[n] = 0$ para $n = 0, 1, \ldots, N/M - 1$.

Partindo da estrutura básica, é possível montar outros tipos de estruturas.

A. Estrutura BFC com Árvore Wavelet

A estrutura BFC (EBFC) para wavelets é mostrada na Figura 4. Para este caso, o código pode ser formado retirando os níveis de detalhes dos primeiros estágios $u_1^{(j)}[n]$. Na verdade não há uma regra específica na escolha dos detalhes a serem retirados; essa escolha influencia diretamente nos parâmetros K e d do código. Retirando os $J_0 - 1$ primeiros detalhes para

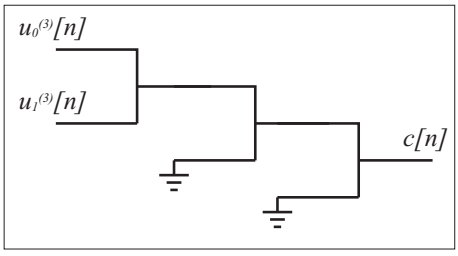

Fig. 4. Estrutura BFC com árvore wavelets

geração de $c[n]$, tem-se

$$c[n] = \sum_{j=J_0}^{J} \sum_{l=0}^{N/2^j-1} u_1^{(j)}[l] g_1^j[n - 2^j l] + \sum_{l=0}^{N/2^J-1} u_0^{(J)}[l] g_0^J[n - 2^J l], \quad (7)$$

onde as seqüências $g_0^{(j)}$ e $g_1^{(j)}$ podem ser computadas através de suas transformadas Z cíclicas , dadas por

$$G_0^{(j)}(z) = \prod_{i=0}^{j-1} G_0(z^{2^i}) = G_0^{(j-1)}(z) G_0(z^{2^{j-1}}) \quad (8)$$

e

$$G_1^{(j)}(z) = G_1(z^{2^{j-1}}) \prod_{i=0}^{j-2} G_0(z^{2^i}) = G_1^{(j-1)}(z) G_1(z^{2^{j-1}}). \quad (9)$$

Para este caso, a decodificação fica

$$\tilde{u}_i^{(j)}[l] = \sum_{n=0}^{N-1} \tilde{c}[n] h_i^{(j)}[2^j l - n], \quad (10)$$

para $l = 0, 1, \ldots, N/2^j - 1$ e $j = J_0, J_0 + 1, \ldots, J$. As síndromes são dadas por

$$s_1^{(j)}[l] = \sum_{n=0}^{N-1} \tilde{c}[n] h_1^{(j)}[2^j l - n], \quad (11)$$

para $l = 0, 1, \ldots, N/2^j - 1$ e $j = 1, 2, \ldots, J_0 - 1$, onde

$$H_0^{(j)}(z) = \prod_{i=0}^{j-1} H_0(z^{2^i}) = H_0^{(j-1)}(z) H_0(z^{2^{j-1}}) \quad (12)$$

e

$$H_1^{(j)}(z) = H_1(z^{2^{j-1}}) \prod_{i=0}^{j-2} H_0(z^{2^i}) = H_1^{(j-1)}(z) H_1(z^{2^{j-1}}). \quad (13)$$

Exemplo 1: Considere a EBFC projetada para $GF(9)$ com $N = 8$. É possível projetar os filtros no corpo $GF(3) \subset GF(9)$, $J = 3$. A estrutura está mostrada na Figura 4 e forma o código $C_1(8, 2, d)$. Os filtros escolhidos, satisfazendo3 a condição de recuperação perfeita (RP) (Apêndice I), são:
$G_0(z) = 2 + 2z^{-1} + 2z^{-5} + 2z^{-6}$,
$G_1(z) = 2z^{-1} + 2z^{-2} + 2z^{-3}$,
$H_0(z) = 1 + 2z^{-1} + z^{-2}$,
$H_1(z) = 2 + 2z^{-4} + z^{-5} + z^{-7}$. Analisando a distância mínima do código para essa estrutura com esses filtros, tem-se $d = 4$. A equação do código se reduz a

$$c[n] = u_1^{(3)}[0] g_1^j[n] + u_0^{(3)}[0] g_0^J[n].$$

Gerando a palavra código a partir de
$u_1^{(3)}[0] = 1$;
$u_0^{(3)}[0] = 2$;
tem-se:
$c[n] = [0\ 2\ 0\ 2\ 1\ 2\ 1\ 2]$. Como $d = 4$, esse código detecta até três erros. A EBFC de análise é mostrada na Figura 5. Considerando
$\tilde{c}[n] = [0\ 2\ 0\ 2\ 1\ 2\ 1\ 2] + [1\ 0\ 0\ 2\ 0\ 1\ 0\ 0]$, o decodificador apresenta
$s_1^{(1)}[n] = [0\ 0\ 0\ 0]$,
$s_1^{(2)}[n] = [1\ 1] \neq 0$, (Erro detectado)
$u_1^{(3)}[0] = 0$,
$u_0^{(3)}[0] = 1$. Modificando a estrutura de entrada da EBFC para anular a posição $u_1^{(3)}[0]$, o código resultante é o de repetição $C(8, 1, 8)$.

O código produzido varia com os filtros projetados e com a estrutura de entrada escolhida, possibilitando uma grande variação de códigos a serem produzidos.

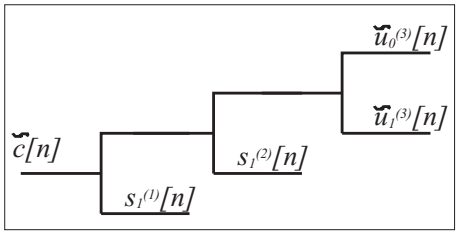

Fig. 5. Estrutura BFC de análise do exemplo 1.

B. Estrutura BFC com Árvore Completa

A EBFC com árvore completa é uma estrutura mais ampla, no sentido de que engloba as estruturas BFC com árvore wavelet. Um exemplo dessa estrutura é apresentada na Figura 6. Nessa estrutura, escolhe-se a EBFC básica e forma-se uma

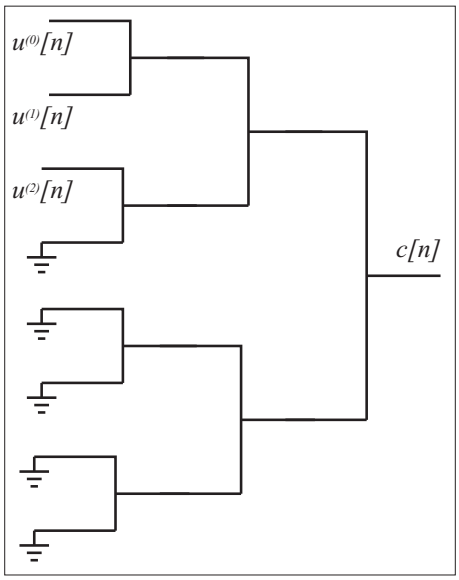

Fig. 6. Estrutura BFC com árvore completa.

árvore completa até o maior número possível de estágios. Escolhe-se a estrutura de entrada e o código é produzido por interações BFC, assim como na estrutura anterior. Para essa situação, a transformada associada é análoga a transformada de Fourier de curta duração (TFCD) [2]. A TFCD sobre corpos finitos (TFCDCF), para um BFC básico de M canais e com J estágios, é dada por

$$x^{(i)}[l] \triangleq \sum_{n=0}^{N-1} x[n] h^{(i)}[M^J l - n], \quad (14)$$

para $l = 0, 1, \ldots, N/M^J - 1$ e $i = 0, 1, \ldots, M^J - 1$. A TFCDCF inversa é dada por

$$x[n] = \sum_{i=0}^{M^J-1} \sum_{l=0}^{N/M^J-1} x^{(i)}[l] g^{(i)}[n - M^J l], \quad (15)$$

para $n = 0, 1, \ldots, N-1$. As seqüências $h^{(i)}$ e $g^{(i)}$ são obtidas no domínio Z por

$$H^{(\sum_{j=0}^{J-1} a_j M^j)}(z) = \prod_{j=0}^{J-1} H_{a_j}(z^{M^{J-1-j}}) \tag{16}$$

e

$$G^{(\sum_{j=0}^{J-1} a_j M^j)}(z) = \prod_{j=0}^{J-1} G_{a_j}(z^{M^{J-1-j}}), \tag{17}$$

onde os a_j podem assumir valores entre 0 e $M-1$. A partir desses valores, obtém-se todos os filtros $G^{(i)}(z)$ e $H^{(i)}(z)$.

Supondo o caso particular em que $M = 2$ e $N = 2^J$, usando a notação para códigos e escolhendo a estrutura de entrada, tem-se

$$c[n] = \sum_{i \in \mathbf{C}} u^{(i)}[0] g^{(i)}[n], \tag{18}$$

sendo as equações de decodificação e de síndrome dadas por

$$\tilde{u}^{(i)}[0] = \sum_{n=0}^{N-1} \tilde{c}[n] h^{(i)}[-n], \tag{19}$$

para $i \in \mathbf{C}$, e

$$s^{(i)}[0] = \sum_{n=0}^{N-1} \tilde{c}[n] h^{(i)}[-n], \tag{20}$$

para $i \notin \mathbf{C}$.

***Exemplo** 2:* Considera-se a mesma EBFC básica do exemplo 1 em $GF(9)$, com a estrutura de entrada mostrada na Figura 7. Nessa situação o código gerado é $C_2(8, 3, 4)$ para $GF(3)$. O código pode ser obtido por

$$c[n] = \sum_{i=0}^{2} u^{(i)}[0] g^{(i)}[n],$$

onde

$$G^{(0)}(z) = G_0(z) G_0(z^2) G_0(z^4),$$
$$G^{(1)}(z) = G_0(z) G_0(z^2) G_1(z^4),$$
$$G^{(2)}(z) = G_0(z) G_1(z^2) G_0(z^4).$$

Para decodificação e cálculo da síndrome, pode-se utilizar a estrutura da Figura 8.

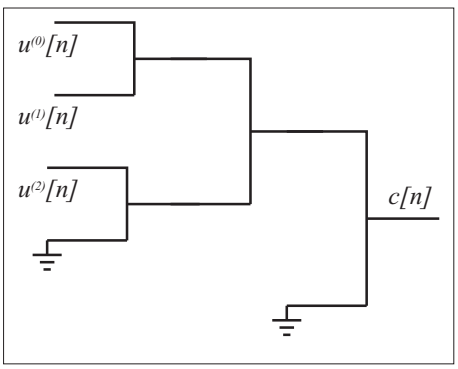

Fig. 7. Estrutura BFC de síntese com árvore completa do exemplo 2.

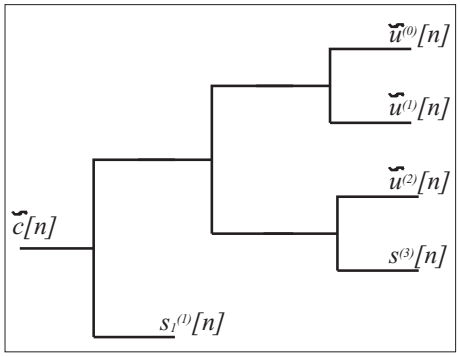

Fig. 8. Estrutura BFC de análise com árvore completa do exemplo 2.

C. Estrutura BFC Mista

Estruturas BFC mistas são formadas por mais de uma EBFC básica com número de canais diferentes. Dessa forma é possível fatorar o comprimento N do código. Um exemplo para $N = 15$ está mostrado na Figura 9.

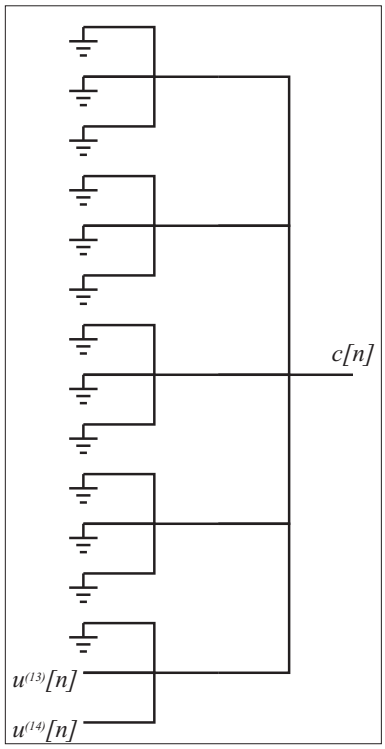

Fig. 9. Estrutura BFC mista para $N = 15$, mesma configuração do exemplo 3.

***Exemplo** 3:* Considere o corpo $GF(2^4)$, com $N = 15$ e α raiz do polinômio primitivo $\pi(x) = 1 + x + x^4$. Os filtros da estrutura básica de 3 canais são dados no domínio da TFCF:
$H_0^3[k] = G_0^3[k] = [1\ 1\ 1\ 1\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]$,
$H_1^3[k] = G_1^3[k] = [0\ 0\ 0\ 0\ 0\ 1\ 1\ 1\ 1\ 1\ 0\ 0\ 0\ 0\ 0]$,
$H_2^3[k] = G_2^3[k] = [0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 1\ 1\ 1]$,
e os da estrutura básica de 5 canais, por
$H_0^5[k] = G_0^5[k] = [1\ 1\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]$,
$H_1^5[k] = G_1^5[k] = [0\ 0\ 0\ 1\ 1\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]$,
$H_2^5[k] = G_2^5[k] = [0\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 1\ 0\ 0\ 0\ 0\ 0\ 0]$,

$H_3^5[k] = G_3^5[k] = [0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 1\ 0\ 0\ 0]$,
$H_4^5[k] = G_4^5[k] = [0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 1]$.
Filtros construídos por essa regra sempre satisfazem a condição RP. A estrutura de entrada da Figura 9 gera o código Reed-Solomon $C_3(15, 2, 14)$ com polinômio gerador dado por $g(x) = (x - \alpha^0)(x - \alpha^1)\ldots(x - \alpha^{12})$.

III. Projeto de Códigos com árvore completa de J-estágios

Nesta seção é apresentado um método para o projeto de bons códigos lineares (no sentido de maximizar d) com estruturas BFC para um dado corpo $GF(q)$. O procedimento consiste nos seguintes passos (método ACJ):

1) Escolher um valor apropriado para $N = M^J$ o qual divide $q^m - 1$, para algum valor de m;
2) Projetar os filtros da estrutura básica BFC. O código muda de acordo com o filtro produto, $P(z) = H_0(z)G_0(z)$, e para cada fatoração escolhida (Apêndice I);
3) Calcular a matriz de transformação da TFCDCF encontrando os $g^{(i)}[n]$ através de (17). Cada coluna dessa matriz constitui as linhas da EBFC com árvore completa. Escolher as componentes de entrada ($i \in \mathbf{C}$) significa escolher as linhas da matriz G geradora do código. A quantidade de colunas escolhidas constitui o parâmetro K do código;
4) Escolher as colunas cuja combinação linear resulte no código de maior distância mínima, definindo a estrutura de entrada. A sugestão é aplicar a TFCF na matriz de transformação da TFCDCF e escolher o grupo que apresentar maior número de zeros consecutivos em suas combinações lineares. Dessa forma, se existe $\delta - 1$ zeros consecutivos, o código terá $d \geq \delta$ (cota BCH [12]).

***Exemplo** 4:* Código de Hamming $C_4(4, 2, 3)$ em $GF(3)$, pelo método ACJ.

1) $N = 4 = 2^2 \Rightarrow 4|(3^2 - 1)$.
2) Escolhe-se o filtro produto $P(z) = 1$ que satisfaz a condição RP. Isto significa que
   $H_0(z) = G_0(z)^{-1}(mod\ z^{-4} - 1)$.
   Escolhendo
   $H_0(z) = z^{-1} + z^{-2} + 2z^{-3}$,
   $G_0(z) = 1 + z^{-1} + 2z^{-3}$,
   $H_1(z) = 2 + 2z^{-1} + z^{-2}$,
   $G_1(z) = 1 + 2z^{-1} + 2z^{-2}$.
3) Calculando todos os $g^{(i)}[n]$, tem-se que

$$\begin{bmatrix} c[0] \\ c[1] \\ c[2] \\ c[3] \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 1 & 1 & 2 & 0 \\ 0 & 2 & 2 & 2 \\ 2 & 2 & 0 & 1 \end{bmatrix} \begin{bmatrix} u^{(0)}[0] \\ u^{(1)}[0] \\ u^{(2)}[0] \\ u^{(3)}[0] \end{bmatrix}.$$

Escolhe-se a segunda e a terceira coluna da matriz de transformação da TFCDCF para definir a estrutura de entrada. A EBFC mostrada na Figura 10 gera o código de Hamming ternário $C_4(4, 2, 3)$. Essa estrutura pode ser simplificada para a estrutura da Figura 11. A estrutura de análise está na Figura 12 e a matriz geradora do código, G, pode ser obtida pela matriz da TFCDFC, resultando em

$$G = \begin{bmatrix} 0 & 1 & 2 & 2 \\ 1 & 2 & 2 & 0 \end{bmatrix}.$$

A matriz de paridade H pode ser encontrada através de (20), sendo dada por

$$H = \begin{bmatrix} 0 & 2 & 1 & 1 \\ 2 & 1 & 1 & 0 \end{bmatrix}.$$

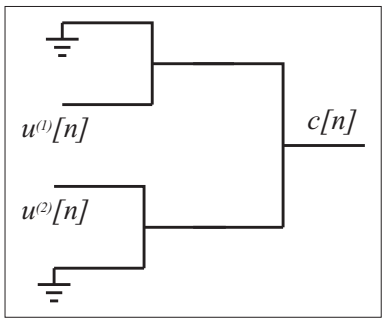

Fig. 10. Estrutura BFC geradora do código de Hamming $C_4(4, 2, 3)$, em $GF(3)$, do exemplo 4.

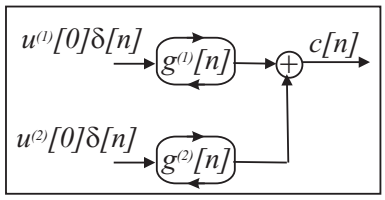

Fig. 11. Estrutura geradora do exemplo 4 simplificada.

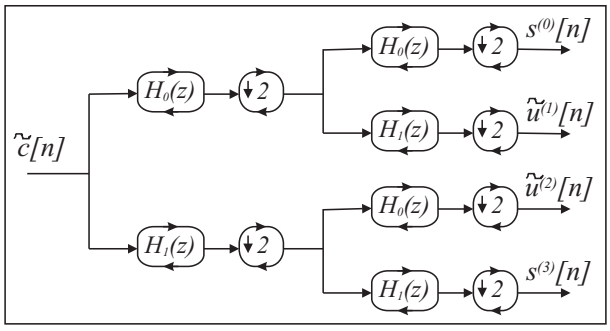

Fig. 12. Estrutura de análise do exemplo 4 simplificada.

***Exemplo** 5:* Em $GF(3)$, procura-se um código linear $C_5(8, 2, d)$ com máximo d, utilizando a mesma estrutura básica BFC do exemplo 1. Pode-se escrever a matriz da TFCDCF como sendo

$$c[n] = \begin{bmatrix} 1 & 1 & 0 & 1 & 2 & 2 & 2 & 0 \\ 1 & 0 & 2 & 0 & 1 & 2 & 0 & 1 \\ 1 & 1 & 0 & 2 & 2 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 2 & 0 & 2 & 2 & 1 & 2 & 0 \\ 1 & 0 & 2 & 0 & 1 & 1 & 0 & 2 \\ 1 & 2 & 0 & 1 & 2 & 0 & 1 & 2 \\ 1 & 0 & 1 & 0 & 1 & 2 & 0 & 2 \end{bmatrix} u^{(i)}[0].$$

Escolhendo as colunas 6 e 8, obtem-se a matriz G do código

$$G = \begin{bmatrix} 2 & 2 & 0 & 1 & 1 & 1 & 0 & 2 \\ 0 & 1 & 1 & 1 & 0 & 2 & 2 & 2 \end{bmatrix}$$

e a EBFC mostrada na Figura 13. Este código possui parâmetros $C_5(8, 2, 6)$.

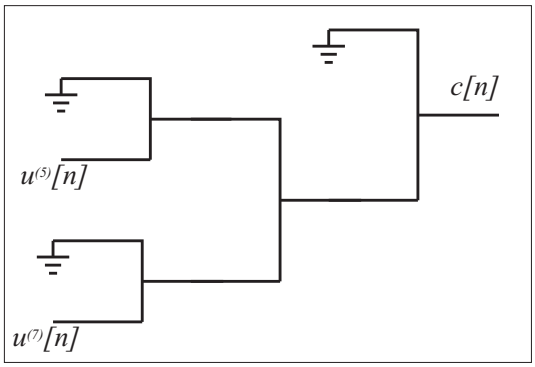

Fig. 13. Estrutura BFC do exemplo 5, $C_5(8, 2, 6)$.

IV. Conclusões

Nesse artigo foi mostrado como se utilizar estruturas de banco de filtros cíclicos (BFC) para codificar e decodificar códigos de bloco lineares. A transformada de Fourier de curta duração sobre corpos finitos foi apresentada e utilizada para propor um método de projeto de bons códigos para uma dada estrutura BFC básica. Foram apresentados exemplos de códigos construídos por esse método, obtendo-se códigos lineares conhecidos, tais como os códigos de Hamming, de repetição e Reed-Solomon. Estruturas BFC apresentam convoluções, compressões e expansões cíclicas, o que significa a possibilidade de fácil implementação em *hardware*. Métodos de correção de erros estão sendo investigados, utilizando-se estruturas BFC de análise.

Referências

[1] A. V. Oppenheim and R. W. Schafer, *Discrete-time Signal Processing*, Prentice Hall, Upper Saddle River, New Jersey, 1998.
[2] G. Strang and T. Nguyen, *Wavelets and Filter Banks*, Wellesley Cambridge, USA, 1997.
[3] M. Vetterli and J. Kovacevic , *Wavelets and Subband Coding*, Prentice Hall, Upper Saddle River, New Jersey, 1995.
[4] R. J. McEliece, *Finite Field for Computer Scientists and Engineers*, KAP, Norwell, Massachusetts, 1987.
[5] J. M. Pollard, *The Fast Fourier Transform in a Finite Field*, Math Comput., Vol. 25, No 114, pp. 365-374, Apr. 1971.
[6] G. Caire and R. L. Grossman, *Wavelet Transforms Associated with Finite Cyclic Groups*, IEEE Transaction on Information Theory, Vol 39, No 4, pp. 1157-1166, July 1993.
[7] T. Cooklev, A. Nishihara and M. Sablatash, *Theory of Filter Banks over Finite Fields*, 1994 IEEE Asia-Pacific Conference on Circuits and Systems, pp. 260-265, Taipei, Taiwan, Dec. 1994.
[8] H. M. de Oliveira, T. H. Falk e R. F. G. Távora, *Decomposição de Wavelets sobre Corpos Finitos*, Revista da Sociedade Brasileira de Telecomunicações, Vol 17, No 1, pp. 38-47, 2002.
[9] R. M. Campello Souza, H. M. de Oliveira e D. Silva, *The Z Transform over Finite Fields*, Proceedings of the International Telecommunications Symposium - ITS2002, (em CD) Natal, Brazil, setembro 2002.
[10] G. J. da Silva Jr. e R. M. Campello Souza, *Banco de Filtros e Wavelets para Sistemas Cíclicos sobre Corpos Finitos*, XXV Simpósio Brasileiro de Telecomunicações - SBrT, Recife, PE, setembro 2007 (aceito para apresentação).
[11] F. Fekri, S. W. McLaughlin, R. M. Mersereau and R. W. Schafer *Block Error Correcting Codes Using Finite-Field Wavelet Transforms*, IEEE Transactions on Signal Processing, Vol. 54, NO. 3, March 2006.
[12] S. Lin and D. J. Costello, Jr., *Error Control Coding*, 2a. ed., Prentice Hall, 2004.
[13] R. E. Blahut, *Theory and Pratice of Error Control Codes*, Addison-Wesley, 1983.
[14] A. J. Menezes, P. C. van Oorschot e S. A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, 1996.

Apêndice I
Sistemas Cíclicos

A. Filtros Cíclicos

O diagrama da Figura 14 denota um filtro ou sistema *cíclico linear invariante no tempo* (CLIT). Um sistema CLIT também pode ser caracterizado por sua resposta ao impulso, $h[n]$, pela relação

$$y[n] = x[n] \circledN h[n], \tag{21}$$

onde $\circledN$ denota convolução cíclica de comprimento N.

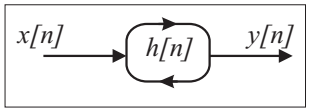

Fig. 14. Diagrama de um sistema CLIT

B. A Transformadas Z Cíclica

Para cada seqüência $x[n]$, $n = 0, 1, \ldots, N-1$, existe um polinômio $X(z)$, com coeficientes em $GF(p^m)$, na variável z^{-1}, o qual representa a transformada Z cíclica de $x[n]$, denotada por

$$x[n] \overset{Z}{\longleftrightarrow} X(z),$$

se

$$X(z) \triangleq \sum_{n=0}^{N-1} x[n] z^{-n}. \tag{22}$$

A transformada Z cíclica possui propriedades sobre a aritmética polinomial módulo $(z^{-N} - 1)$.

C. Projeto de Banco de Filtros Cíclicos

Um método simples para projeto de banco de filtros cíclicos é apresentado a seguir:

- Escolher um filtro produto, $P(z)$, satisfazendo a equação

$$P(z) + P(-z) = 2. \tag{23}$$

- Fatorar $P(z)$ em $H_0(z)G_0(z)$;
- Utilizar as seguintes equações para encontrar $H_1(z)$ e $G_1(z)$:

$$H_1(z) = -z^l G_0(-z) \tag{24}$$

e

$$G_1(z) = -z^{-l} H_0(-z), \tag{25}$$

com l ímpar. Diz-se que esses filtros satisfazem a condição de recuperação perfeita (RP).