CRITICALLY SUBSAMPLED FILTERBANKS IMPLEMENTING REED-SOLOMON CODES

Geert Van Meerbergen ⋆, Marc Moonen ⋆, Hugo De Man ⋆†

⋆ E.E. Dept., ESAT/SISTA, K.U.Leuven Kasteelpark Arenberg 10, 3001 Leuven, Belgium
† IMEC, Kapeldreef 75, 3001 Leuven, Belgium
gvanmeer,moonen@esat.kuleuven.ac.be, deman@imec.be

ABSTRACT

The last decade shows a growing interest in soft decoding techniques. These techniques are almost never applied to existing nearly perfect codes, but instead other families of concatenated codes arise with Turbo codes as the main example. The problem is that for the application of soft decoding, the perfect codes need to be broken into several smaller component codes. In this paper, the family of Reed-Solomon Codes is considered and using filterbanks, they are broken into several component codes (one in each subband). This paper focuses on the construction of such filterbanks, and gradually evolves towards a critically subsampled filterbank. The critical subsampling is crucial if the filterbank is going to be used in a soft decoding setup.

1. INTRODUCTION

In [1], it is shown that many famous nearly perfect codes have a Toeplitz or quasi-cyclic structure, with *Reed-Solomon Codes* (RS), BCH and QR codes as the main examples. These codes can be convolutionally encoded and classical coding theory then gives us the tools to decode some of them, albeit without using soft information. In the last decade however, there is an emerging trend towards soft error correcting decoding. The convolutional structure of the codes can seemingly be exploited in *BCJR's algorithm* [2], however the constraint length of many good codes is too large to get an algorithm that is feasible from a complexity point of view. Therefore, another family of codes was developed, that can be soft decoded. Most of these codes are concatenated codes, as introduced by Forney [3], that are built up from several smaller codes, thereby reducing decoding complexity. These codes are not perfect anymore, especially so for smaller block lengths, and Turbo codes serve as the main example. In this paper, instead of concatenating codes, a perfect code (e.g. RS) is broken down by means of a filterbank. We will only consider RS codes in this paper, as they are commonly used in many telecommunication systems. In a first section, it is explained how a filterbank with one tap subband filters can be built based on *Cook Toom's algorithm*. In section 3, a filterbank is constructed that has non-zero order subband filters (encoders). Because the filterbank is not critically downsampled, it can not be used in a soft decoding algorithm, and therefore a critically downsampled bank is built in section 4. Finally, conclusions are drawn.

2. COOK-TOOM'S ALGORITHM AS THE BASIS OF A FILTERBANK IMPLEMENTATION

Straightforwardly computing a product $y(z^{-1})$ of two polynomials $g(z^{-1}) = \mathbf{g}_0 + \mathbf{g}_1 z^{-1} + ... + \mathbf{g}_{L-1} z^{-L+1}$ and $u(z^{-1}) = \mathbf{u}_0 + \mathbf{u}_1 z^{-1} + ... + \mathbf{u}_{N-1} z^{-N+1}$ requires NL multiplications. The application of *Cook-Toom*'s algorithm [4] is known to reduce the number of multiplications to $M \geq N + L - 1$. The procedure is as follows: First, choose a set of interpolation points $\{r_i\}_{i=0..M-1}$ that are the roots of $R(z^{-1}) = \prod_{i=0}^{M-1} (z^{-1} - r_i)$. Evaluate $y(r_i) = g(r_i)u(r_i)$ and perform *Lagrange* interpolation $y(z^{-1}) = \sum_{i=0}^{M-1} y(r_i) L_i(z^{-1})$ with $L_i(z^{-1}) = \frac{\prod_{k \neq i} (z^{-1} - r_k)}{\prod_{k \neq i} (r_i - r_k)}$.

Example 1 As an example, the product of $g(z^{-1}) = \mathbf{g}_0 + \mathbf{g}_1 z^{-1}$ $(L = 2)$ and $u(z^{-1}) = \mathbf{u}_0 + \mathbf{u}_1 z^{-1}$ $(N = 2)$ is calculated. With the interpolation points chosen to be $\{0, 1, -1\}$, $y(0) = \mathbf{u}_0 \mathbf{g}_0$, $y(1) = (\mathbf{u}_0 + \mathbf{u}_1)(\mathbf{g}_0 + \mathbf{g}_1)$, $y(-1) = (\mathbf{u}_0 - \mathbf{u}_1)(\mathbf{g}_0 - \mathbf{g}_1)$. Finally, the *Lagrange* polynomials are calculated $L_0(z^{-1}) = (1 - z^{-2})$, $L_1(z^{-1}) = (z^{-1} + z^{-2})/2$, $L_{-1}(z^{-1}) = (-z^{-1} + z^{-2})/2$, and $y(z^{-1}) = y(0)L_0(z^{-1}) + y(1)L_1(z^{-1}) + y(-1)L_{-1}(z^{-1})$ is reconstructed. □

If the polynomial multiplication is written as $\mathbf{y} = \mathbf{G}\mathbf{u}$, then *Cook-Toom*'s algorithm can be viewed as a matrix decomposition $\mathbf{G} = \mathbf{C}\text{diag}(\mathbf{B}\mathbf{g})\mathbf{A}$, with $\mathbf{G}$ the $(N + L - 1) \times N$ Toeplitz matrix defining the filter $g(z^{-1})$. $\mathbf{A}$ is an $M \times N$ matrix with $\mathbf{A}_{m,n} = r_m^n (n = 0..N-1)$, $\mathbf{B}$ is an $M \times L$ matrix with $\mathbf{B}_{m,l} = r_m^l (l = 0..L-1)$ and $\mathbf{C}$ is an $(N + L - 1) \times M$ with the ith column containing the first $N + L - 1$ coefficients of $L_i(z^{-1})$.

Example 2 Continuing example 1, we obtain:

$$\mathbf{y} = \underbrace{\begin{bmatrix} \mathbf{g}_0 & \\ \mathbf{g}_1 & \mathbf{g}_0 \\ & \mathbf{g}_1 \end{bmatrix}}_{\mathbf{G}} \mathbf{u} = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{2} & -\frac{1}{2} \\ -1 & \frac{1}{2} & \frac{1}{2} \end{bmatrix}}_{\mathbf{C}} \mathbf{D} \underbrace{\begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 1 & -1 \end{bmatrix}}_{\mathbf{A}} \mathbf{u} \quad (1)$$

$$\mathbf{D} = \text{diag}\left(\underbrace{\begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 1 & -1 \end{bmatrix}}_{\mathbf{B}} \begin{bmatrix} \mathbf{g}_0 \\ \mathbf{g}_1 \end{bmatrix} \right) \quad (2)$$

□

Note that the same $\mathbf{y}$ *can be obtained by multiplying* $\mathbf{g}_\delta(z^{-1}) = z^{-\delta} g(z^{-1})$ *and* $\mathbf{u}_\delta(z^{-1}) = z^{\delta} u(z^{-1})$. *This results in a somewhat different matrix decomposition where* $\mathbf{A}_{m,n} = r_m^{n+\delta}$ *and* $\mathbf{B}_{m,l} = r_m^{l-\delta}$. *This will be used at the end of the paper. Unless mentioned otherwise,* $\delta = 0$.

Because of the underlying polynomial arithmetic, *Cook-Toom*'s algorithm is also applicable in other fields than the complex field $\mathbb{C}$, which is important if it is used in a coding context where finite fields like the *Galois Field* (GF) are common. Also note that *Cook-Toom*'s algorithm is a special case of the *Chinese Remainder Theorem* [4], and that if the degree of $R(z^{-1})$ is too low $(M < L + N - 1)$, the convolution modulo $R(z^{-1})$ is calculated.

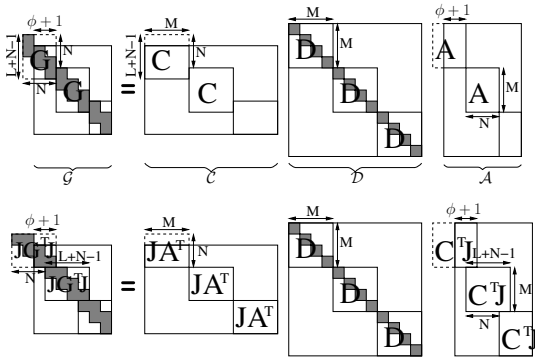

Fig. 1. Filterbank from a matrix decomposition point of view. The *overlap-add* version resp. *overlap-save* is shown in the upper resp. lower part of the figure. The phase ϕ of the downsamplers is also indicated.

The next theorem shows how a filterbank can be constructed starting from the above matrix decomposition.

Theorem 1 *Given is a* $\mathbf{CDA}$ *decomposition based on Cook-Toom implementing an* N *by* L *convolution with* $g(z^{-1})$ *using* $M \geq L+N-1$ *roots. Now, construct a multirate filterbank with* M *bands, indexed by* m*, and subsampled by* N*. The analysis respectively synthesis filters of band* $m = 0..M-1$ *are defined as follows:*

$$h_m(z^{-1}) = \sum_{k=0}^{N-1} \mathbf{A}_{m,N-1-k} z^{-k} \quad (3)$$

$$g_m(z^{-1}) = \sum_{k=0}^{L+N-2} \mathbf{C}_{k,m} z^{-k} \quad (4)$$

The diagonal elements of $\mathbf{D}$ *reside in the subband filters. This filterbank implements an overlap-add convolution with* $z^{-N+1} g(z^{-1})$.

The *overlap-add* scheme is widely described in literature [5] [6]. The filterbank splits a large data-block in small blocks of length N. For each block, the convolution will be calculated, based on the decomposition $\mathbf{CDA}$. This is illustrated in the upper part of **Figure 1**, where the convolutions performed in the analysis bank are represented by a multiplication with $\mathbf{A}$. In the subbands, the multiplication with $\mathbf{D}$ is performed and finally, in the synthesis bank, the overlapping parts of length $L-1$ are added, which explains the name. One can see that from a matrix decomposition point of view, the large Toeplitz matrix is broken into smaller, tall Toeplitz matrices that are each decomposed using *Cook-Toom's algorithm*.

Note: One can also apply signal flow transposition to get the overlap-save variant. Now, the large Toeplitz matrix is broken into fat Toeplitz matrices that are equal to $\mathbf{J}\mathbf{G}^T\mathbf{J}$ *with* $\mathbf{J}$ *the exchange matrix (ones on the anti-diagonal). See also the lower part of* **Figure 1**. *The* $\mathbf{CDA}$ *decomposition is transposed, and is left and right-multiplied with* $\mathbf{J}$*. The resulting (overlap-save) filterbank is exactly the same as its overlap add equivalent, but with synthesis and analysis bank swapped.*

Example 3 In this example, the $RS(15,10,6)$ code [7] is introduced, which is also used throughout the rest of this paper. This linear code in $GF(2^4)$[1] encodes a data word $u(z^{-1})$ of 10 symbols into a codeword $y(z^{-1})$ of 15 symbols by filtering with $g_{RS}(z^{-1}) = \prod_{k=0}^{5}(z^{-1}-\alpha^k)$ $(L=6)$. $L=6$ is also called the dimension D of the code. Let us choose the downsampling factor $N=3$, which then also corresponds to the block length N; Now, take $M = 8 \geq L+N-1$. As roots of $R(z^{-1})$, we pick the first 6 elements of the Galois Field: $r_i = i, i = 0..5$. The resulting *overlap-add* filterbank is shown in **Figure 2**. Note that every codeword $y(z^{-1})$ belongs to a linear subspace $\mathcal{S}_{RS}$ spanned by $z^{-k} g_{RS}(z^{-1}), k = 0..9$, and that the coefficients of these polynomials form the columns of $\mathcal{G}_{\mathbf{RS}}$, which is called the generator matrix of the code. □

[1]The primitive polynomial is x^4+x+1, with $\alpha = 2$ a primitive 15th root of unity. (Decimal notation is used to represent elements of a Galois Field)

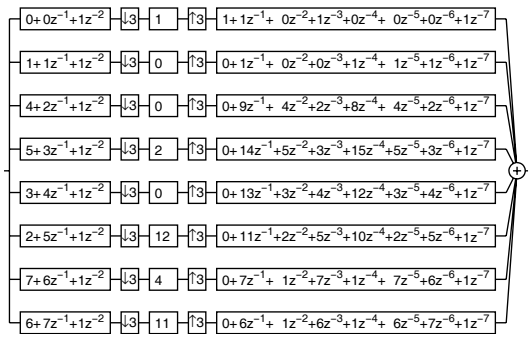

Fig. 2. *overlap-add* filterbank $(L=6, M=8, N=3)$ for the $RS(15,10,6)$ code described in example 3. A dataword of 10 symbols padded with zeros is fed in the filterbank, resulting in a codeword of 15 symbols. Because there are no filters present in the subbands, this bank has no error correcting capacity.

Notice that one large convolution is now broken into 8 pointwise multiplications (zero-order filters) in the subbands, which however themselves do not have error correcting capacity. I.e., non-zero order subband *filters* are needed that work as a convolutional code and that are able to correct errors by introducing redundancy, in this case by zero padding the input. In the next section, a filterbank with component codes in the subbands will be constructed.

3. FILTERBANK WITH COMPONENT CODES IN SUBBANDS

Imagine now that the elements $\mathbf{g}_0, ..., \mathbf{g}_{L-1}$ of the vector $\mathbf{g}$ are a function of z^{-N}, such that $\mathbf{g}_l(z^{-N}) = \sum_{k=0}^{K-1} \mathbf{g}_l[k] z^{-kN}$. Following *Cook-Toom*'s decomposition, the diagonal elements of $\mathbf{D}$ will also be functions of z^{-N}, resulting in filters $\mathbf{d}_0(z^{-1})$, ..., $\mathbf{d}_{L-1}(z^{-1})$ in the subbands. Relying on the superposition principle (see [8] for a proof), it can be shown that the filterbank implements a convolution with $g(z^{-1}) = z^{-N+1} \sum_{l=0}^{L-1} z^{-l} \mathbf{g}_l(z^{-N})$. This can also be written in matrix notation as follows[2]:

$$g(z^{-1}) = \left[z^{-(0:1:L-1)}\right] \cdot \underbrace{\begin{bmatrix} \mathbf{g}_0[0] & \mathbf{g}_0[1] & \dots & \mathbf{g}_0[K-1] \\ \mathbf{g}_1[0] & \mathbf{g}_1[1] & \dots & \mathbf{g}_1[K-1] \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{g}_{L-1}[0] & \mathbf{g}_{L-1}[1] & \dots & \mathbf{g}_{L-1}[K-1] \end{bmatrix}}_{\mathbf{\Gamma}} \cdot \left[z^{-(0:1:K-1)N}\right]^T \quad (5)$$

[2]The following Matlab notation is used $z^{a:b:c} = [z^a z^{a+b} z^{a+2b} \dots z^c]$, if b is not given, $b=1$ is assumed

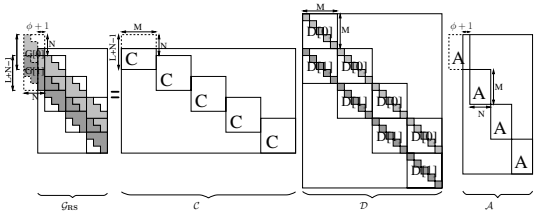

Fig. 3. Matrix decomposition for the RS(15,10,6) filterbank in example 4. The parameters are $K = 2$, $L = N = 3$, $M = 5$, $\phi = 0$.

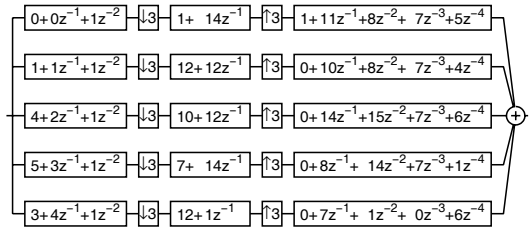

Fig. 4. Filterbank with component codes in each subband for the RS(15,10,6). (example 4)

Note that $K = 1$ results in the filterbank described in section 2. Note also that not every filter can be realized if $L < N$ (leading to missing powers of z^{-1} in equation 5), therefore we impose the constraint $L \geq N$.

Example 4 We obtain a realization of RS(15,10,6) by making the following consecutive choices. Starting with $N = 3$, we choose the smallest possible values for $L(L \geq N)$ and $M(M \geq L + N - 1)$, i.e. $L = 3$ and $M = 5$. The 5 first elements of the GF: $\{0..4\}$ are used as roots of $R(z^{-1})$. Knowing that the filter g_{RS} has 6 coefficients, the matrix $\mathbf{\Gamma}$ is 3×2 $(K = 2)$ as shown below.

$$\begin{bmatrix} 1 & z^{-1} & z^{-2} \end{bmatrix} \underbrace{\begin{bmatrix} 1 & 14 \\ 3 & 3 \\ 14 & 1 \end{bmatrix}}_{\mathbf{\Gamma}} \begin{bmatrix} 1 \\ z^{-3} \end{bmatrix} \tag{6}$$

The resulting filterbank can be found in **Figure 4** and generates the same code subspace $\mathcal{S}_{RS}$ as in Example 3. The corresponding matrix factorization is shown in **Figure 3** where the Toeplitz generator matrix $\mathcal{G}_{\mathbf{RS}}$ is shown. □

Remark that if $L = N$, which is the case in the previous example, this filterbank can be found from the polyphase components of $\mathbf{u}$, $\mathbf{g}$ and $\mathbf{y}$ by applying an aperiodic convolution algorithm as described in [8].

As a second remark, we note that for $M = 5$, an element of order 5 exists in $GF(2^4)$, namely $\beta = \alpha^3$. This element generates a multiplicative subgroup $\beta^0 = 1$, $\beta^1 = 8$, $\beta^2 = 12$, $\beta^3 = 10$, $\beta^4 = 15$, $(\beta^5 = 1)$ that can be used as the roots of $R(z^{-1})$. The filterbank so obtained is shown in **Figure 5**. Looking into more detail, the analysis and synthesis bank calculate the *Mattson-Solomon Polynomial* [7] of their inputs, also known as the DFT in the GF. As will be seen further, this property is very important. The filterbank in **Figure 5** can thus be seen as *the (GF) counterpart of the STFT bank*, which was introduced by [9] [5].

However, the aim of this paper is *not* to simplify the encoding for the RS code (which is not overly complex). We are aiming

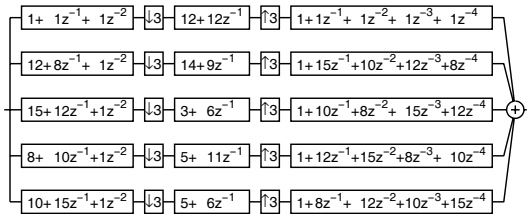

Fig. 5. STFT-like filterbank based on the *Mattson-Solomon Polynomial*

at simplifying the decoding, and for this purpose, the filterbank has to be run in reverse order, i.e. the subband samples must be found starting from the codeword (the output of the filterbank). Because of the fact that the bank is not critically downsampled, the synthesis bank implements a kind of projection from a high dimensional space to a low(er) dimensional one, and hence the subband samples, even in the error free case can never be found. In the following section, this problem is tackled.

4. CRITICALLY DOWNSAMPLED FILTERBANK

In this section, a critically downsampled filterbank is built for the family of RS codes. In order to achieve this goal, the constraint $M \geq L + N - 1$ must be violated $(M = N)$, resulting in a filterbank that does *not* implement a convolution with $g_{RS}(z^{-1})$ but a convolution modulo $R(z^{-1})$. We will try to obtain a circulant convolution, for reasons that will become clear, and therefore choose $R(z^{-1}) = 1 + z^{-M}$. A decomposition of $R(z^{-1})$ in first degree polynomials only exists if M divides $Q = 2^q - 1$. In that case, a subgroup generated by $\beta = \alpha^{Q/M}$ can be found and $R(z^{-1}) = 1 + z^{-M} = \prod_{k=0}^{M-1}(z^{-1} + \beta^k)$ (Fermat's theorem). Hence, the resulting filterbank is based on the *Mattson-Solomon Polynomial*. In the following, a $\mathbf{\Gamma}$ is determined in such a way that the code is still a RS code, however the set of codewords and thus the subspace will be different from the one generated by $\mathcal{G}_{\mathbf{RS}}$. Therefore, properties concerning the code generated by the filterbank are indicated with a superscript IL (Interleaved). Every codeword $y^{IL}(z^{-1})$ obtained as the output of this critically downsampled $(L = M = N)$ bank can be written as $\mathcal{G}^{\mathbf{IL}}_{\mathbf{RS}}\mathbf{u}$. Therefore, the code subspace is spanned by the following set of polynomials:

$$\mathcal{S}^{IL}_{RS} : \left[z^{-(l:l+L-1)}\right] \mathbf{\Gamma} \left[z^{-(k:k+K-1)N}\right]^T \bmod (1 + z^{-N}) \tag{7}$$

with $k = 0 : K-1, l = 0 : L-1$. Stacking the polyphase components (thereby de-interleaving the basis vectors and thus keeping the 'perfect' distance properties of the RS code) gives:

$$\left[z^{-(l:1:l+L-1)Q/N}\right] \mathbf{\Gamma} \left[z^{-(k:1:k+K-1)}\right]^T \bmod (1 + z^{-Q}) \tag{8}$$

with $k = 0 : K-1, l = 0 : L-1$. Now, a $\mathbf{\Gamma}$ can be found such that this set of basis vectors generates the original subspace $\mathcal{S}_{RS}$ defined by $\mathcal{G}_{\mathbf{RS}}$. For $k = l = 0$, a unique $\mathbf{\Gamma}$ can be obtained with only D nonzero elements, including one chosen constant (RS code with dimension D) such that this codeword is in the subspace generated by $\mathcal{G}_{\mathbf{RS}}$. Furthermore, note that for k and/or l nonzero, a shifted version of this codeword $\bmod(1 + z^{-Q})$ is obtained, and therefore is also a codeword (RS code is a cyclic code). Finally, note that $\mathbf{y}$ is obtained using a linear combination of these codewords (RS code is a linear code) and that the critically downsampled filterbank produces the same codewords, but interleaved.

Example 5 To build a critically downsampled filterbank for the RS(15,10,6), let $K = 2$ and $L = M = N = 5$. We choose

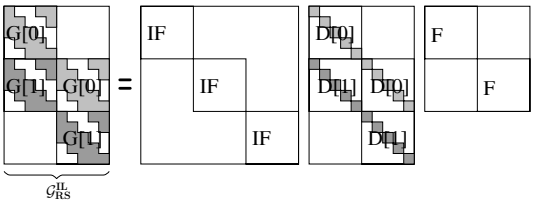

Fig. 6. Matrix decomposition for the critically downsampled ($L = M = N = 5, \phi = 4$) filterbank for $RS(15, 10, 6)$ code, found in example 5, that can be used for soft decoding.

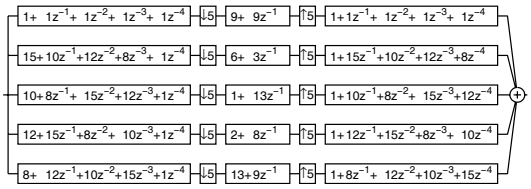

Fig. 7. Critically downsampled ($L = M = N = 5, \phi = 4$) filterbank for $RS(15, 10, 6)$ code, found in example 5

the 2 last rows of $\mathbf{\Gamma}$ to be 0, and $\mathbf{\Gamma}_{0,0}$ to be one. The remaining 5 elements are calculated so that $y(z^{-1})$ is in the subspace generated by $\mathcal{G}_{\mathbf{RS}}$. This gives the unique codeword

$$y_(z^{-1}) = \begin{bmatrix} z^{-0} & z^{-3} & z^{-6} & z^{-9} & z^{-12} \end{bmatrix} \begin{bmatrix} 1 & 6 \\ 14 & 14 \\ 6 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} z^{-0} \\ z^{-1} \end{bmatrix} \quad (9)$$

Note that every shifted version $z^{-k-lQ/N}$, for $k = 0 : 1$ and $l = 0 : 4$ is also a codeword and belongs to $\mathcal{S}_{RS}$. Interleaving this codeword, we obtain

$$y^{IL}(z^{-1}) = \begin{bmatrix} z^{-0} & z^{-1} & z^{-2} & z^{-3} & z^{-4} \end{bmatrix} \begin{bmatrix} 1 & 6 \\ 14 & 14 \\ 6 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} z^{-0} \\ z^{-5} \end{bmatrix} \quad (10)$$

Shifting this codeword as described by equation 8 results in the $\mathcal{G}^{\mathbf{IL}}_{\mathbf{RS}}$ in **Figure 6**. The corresponding filterbank is shown in **Figure 7**. The 2 decompositions based on *Cook-Toom's algorithm* used to construct this filterbank are shown below:

$$\underbrace{\begin{bmatrix} 1 & 0 & 0 & 6 & 14 \\ 14 & 1 & 0 & 0 & 6 \\ 6 & 14 & 1 & 0 & 0 \\ 0 & 6 & 14 & 1 & 0 \\ 0 & 0 & 6 & 14 & 1 \end{bmatrix}}_{\mathbf{G}[0]} = \underbrace{\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 15 & 10 & 12 & 8 \\ 1 & 10 & 8 & 15 & 12 \\ 1 & 12 & 15 & 8 & 10 \\ 1 & 8 & 12 & 10 & 15 \end{bmatrix}}_{\mathbf{IF}} \underbrace{\begin{bmatrix} 9 & 0 & 0 & 0 & 0 \\ 0 & 6 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 13 \end{bmatrix}}_{\mathbf{D}[0]} \underbrace{\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 8 & 12 & 10 & 15 \\ 1 & 12 & 15 & 8 & 10 \\ 1 & 10 & 8 & 15 & 12 \\ 1 & 15 & 10 & 12 & 8 \end{bmatrix}}_{\mathbf{F}}$$

$$\underbrace{\begin{bmatrix} 6 & 0 & 0 & 1 & 14 \\ 14 & 6 & 0 & 0 & 1 \\ 1 & 14 & 6 & 0 & 0 \\ 0 & 1 & 14 & 6 & 0 \\ 0 & 0 & 1 & 14 & 6 \end{bmatrix}}_{\mathbf{G}[1]} = \underbrace{\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 15 & 10 & 12 & 8 \\ 1 & 10 & 8 & 15 & 12 \\ 1 & 12 & 15 & 8 & 10 \\ 1 & 8 & 12 & 10 & 15 \end{bmatrix}}_{\mathbf{IF}} \underbrace{\begin{bmatrix} 9 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 13 & 0 & 0 \\ 0 & 0 & 0 & 8 & 0 \\ 0 & 0 & 0 & 0 & 9 \end{bmatrix}}_{\mathbf{D}[1]} \underbrace{\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 8 & 12 & 10 & 15 \\ 1 & 12 & 15 & 8 & 10 \\ 1 & 10 & 8 & 15 & 12 \\ 1 & 15 & 10 & 12 & 8 \end{bmatrix}}_{\mathbf{F}}$$

$$\mathbf{D}[0] = \text{diag}(\mathbf{F}\mathbf{\Gamma}(:, 0))$$
$$\mathbf{D}[1] = \text{diag}(\mathbf{F}\mathbf{\Gamma}(:, 1))$$

□

Remark that Solomon discovered a circulant structure in the RS codes [1], however this differs from the block Toeplitz structure with circulant blocks derived here which is indeed a filterbank-like structure. As a final remark, note that the overlap-add variant equals the overlap-save filterbank, if $\delta = 1$ in *Cook-Toom's algorithm*. Doing so, the DFT matrix $\mathbf{F}$ of the analysis bank is circularly shifted one step to the right resulting in $\mathbf{C} = \mathbf{J}\mathbf{A}^T$ (See also **Figure 1**). Then, signal flow transposition on this filterbank gives us the same filterbank.

5. CONCLUSION

In this paper, it is explained how a critically subsampled filterbank is built that breaks down a RS code into convolutional component codes. In a first step, *overlap-add* and *overlap-save* filterbanks are reviewed, that operate in the GF based on *Cook-Toom*'s algorithm. These filterbanks have zero order tap filters in their subbands. To allow error correction, in a second step, a filterbank is built that has non-zero order subband filters using the superposition principle. The equivalent of the STFT bank is presented, making use of the *Mattson-Solomon polynomial*. However, the latter cannot be used for decoding as it is not critically downsampled. In a third step, it is explained how to build a critically downsampled filterbank, which is remarkable noting that a critically downsampled bank can never perform linear filtering operations. It is again based on the *Mattson-Solomon polynomial* or DFT in the GF and on the fact that it diagonalizes a circulant matrix. In an upcoming paper, it is explained how the component codes are working together in an iterative algorithm to do soft decoding.

Acknowledgments

Geert Van Meerbergen is a Research Assistant with the F.W.O. Vlaanderen. This research work was carried out at the ESAT laboratory of the Katholieke Universiteit Leuven, in the frame of the Interuniversity Poles of Attraction Programme P5/22 and P5/11, the Concerted Research Action GOA-MEFISTO-666, Research Project FWO nr.G.0196.02. The scientific responsibility is assumed by its authors.

6. REFERENCES

[1] G. Solomon and H. C. A. van Tilborg, "A connection between block and convolutional codes," *SIAM J. Appl. Math*, vol. 37, no. 2, pp. 358–369, Oct. 1979.

[2] F. L. Bahl, J. Cocke and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *Information Theory, IEEE Transactions on*, vol. 20, no. 2, pp. 284 –287, Mar. 1974.

[3] G.D. Forney, Jr., *Concatenated Codes*. M.I.T. Press, 1966.

[4] R.E. Blahut, *Fast algorithms for Digital Signal Processing*. Reading, MA: Addison-Wesley, 1984.

[5] M. R. Portnoff, "Time-frequency representation of digital signals and systems based on short-time fourier analysis," *IEEE Transactions on Acoustic, Speech, and Signal Processing*, vol. 35, pp. 356–372, Mar. 1987.

[6] M. Vetterli and J. Kovacevi'c, *Wavelets and Subband Coding*. Englewood Cliffs, NJ: Prentice Hall, 1995.

[7] N.J.A. Sloane and F.J. MacWilliams, *The Theory of ErrorCorrecting Codes*. North-Holland, Amsterdam, 1977.

[8] M. Vetterli, "Running FIR and IIR filtering using multirate filter banks," *Acoustics, Speech, and Signal Processing, IEEE Transactions on*, vol. 36, no. 5, pp. 730 –738, May 1988.

[9] R.E. Crochiere and L.R. Rabiner, *Multirate Digital Signal Processing*. Englewood Cliffs,NJ: Prentice-Hall, 1983.