Soft-In Soft-Out Reed-Solomon Decoding using Critically Subsampled Filterbanks

Geert Van Meerbergen ⋆, Marc Moonen ⋆
⋆ E.E. Dept., ESAT/SISTA, K.U.Leuven
Kasteelpark Arenberg 10, 3001 Leuven-Heverlee, Belgium
Email: {gvanmeer,moonen}@esat.kuleuven.ac.be
Tel.: +32-16-32-{1817,1060}

Hugo De Man ⋆†
† IMEC,
Kapeldreef 75, 3001 Leuven, Belgium
Email: deman@imec.be
Tel.: +32-16-28-1200

***Abstract*— In the last decade, there has been a growing interest in soft decoding techniques. These techniques are used in the context of concatenated codes, with Turbo codes as the main example, but are almost never applied to existing classical codes. In this paper, the family of Reed-Solomon codes is considered and the complexity problem of Soft-In Soft-Out RS decoding is tackled by breaking RS codes into several smaller subcodes. The decomposition, presented here, is based on filterbanks with one subcode in each subband. The critical subsampling is then crucial if the filterbank is going to be used in a soft decoding setup. This paper focuses on the construction of such filterbanks, starting from non-critically subsampled filterbanks and evolving to a critically subsampled filterbank.**

I. Introduction

The importance of Reed-Solomon codes (RS) [1] nowadays can hardly be underestimated. Their application domain spans from storage devices (including magnetic tape, CD, DVD, barcodes, etc) over high-speed modems (such as ADSL, xDSL, etc), wireless and mobile communications (including cellular telephones, microwave links, etc) to satellite and deep space communications. In most of these applications, algebraic decoding algorithms for hard-decision decoding are used, e.g. Berlekamp Massey's algorithm and Guruswami-Sudan's list decoding algorithm [2]. The potential benefit of soft decision RS decoding is rougly 2dB, as discussed in [3]. Thus in light of the ubiquity of RS codes, and taking into account the possible soft decoding gain, soft decoding of RS codes is currently one of the most important problems in coding theory.

Soft decoding of RS (and other powerfull block codes) is however much more complex, as can be illustrated by a short overview of algorithms published in the literature. The trellis based schemes of Hagenauer [4] show a good performance, but become infeasible from a complexity point of view when applied to RS codes. The same holds for Fossorier and Lin's algorithm [5], based on ordered statistics. An alternative approach to the problem of efficient soft decoding is the generalized minimum distance decoding (GMD) of Forney [6]. While the complexity of GMD decoding is moderate, the gain that can be realized is also moderate. Recently, Koetter and Vardy developed a soft front-end for hard RS decoding in [7], resulting in a Soft In Hard Out algorithm. However, Benedetto, Divsalar and Hagenauer describe in [8] that Soft In Soft Out (SISO) is required since it allows soft information to be exchanged between the different blocks in the receiver. This iterative strategy is called the *Turbo Principle*, which is the basic mechanism behind joint equalisation, demodulation and channel decoding. What is needed is an efficient SISO RS decoder.

Recently, this problem of efficient SISO decoding was solved for concatenated codes. The subcodes are then decoded one after the other, thereby reducing the complexity. Concatenated coding was first introduced for block codes by Forney [9] but has indeed attracted a lot of attention recently when SISO decoding became possible for the subcodes. First, convolutional codes were taken as subcodes that could be SISO decoded using BCJR's algorithm [10] resulting in Turbo codes [11]. Recent work by Lodge and Hagenauer [4] has shown good performance with small block codes as subcodes too. At the top level however, there is still little control over the properties of the overall code, e.g. the minimum distance between the codewords.

In this paper, however, the opposite approach is taken. Starting from an overall RS code, a top-down decomposition is developed into many smaller subcodes. Note that Vardy [12] also identifies subcodes for a RS code, which are however operating in the binary field (subfield subcodes). In this paper, specific properties - such as the cyclic nature of RS codes - are exploited to unveil a critically subsampled filterbank structure behind RS codes. The focus of this contribution is on the filterbank aspects.

It is certainly not the first time that filterbanks appear in communication systems, although filterbanks have been known to be a powerfull tool for image and audio processing for a long time now. Recent work by Scaglione *et al.* [13] shows that many modulation schemes including OFDM, DMT, TDMA, and CDMA can actually be viewed as filterbanks that build input diversity (add redundancy) at the transmitter. Filterbanks that add redundancy with the purpose of error correction - and therefore work in finite fields - were also addressed by Fekri [14]. However the filterbanks presented in this paper show a completely different structure and implement powerfull RS codes.

II. Filterbanks and Linear time invariant systems

Filterbanks and multirate systems in general are widely used in digital signal processing. Essentially, these systems work block oriented, i.e. the data is divided in blocks of N (with N the subsampling) and is processed accordingly. More precisely, filterbanks which calculate the *Discrete Fourier Transform*

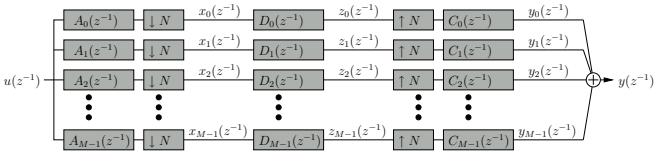

Fig. 1. *overlap-add* filterbank with M bands and N-fold subsampling.

(DFT) of subsequent data blocks are said to be based on the STFT. In this section, some basic facts of STFT filterbanks are recalled to provide a clear understanding of the rest of the paper. Since error correcting codes in the Galois Field $GF(p^d)$ are targeted, this opportunity will be used to present the Galois Field counterpart of STFT filterbanks in the complex field. In this context, $\alpha \in GF(p^d)$ represents a (p^d-1)-th primitive root of unity. The $M \times M$ DFT matrix is then defined by $\mathbf{F}_{ij} = \beta^{ij}$ with $i,j = 0..M-1$ and β an M-th root of unity. This only exists if M divides p^d-1, in which case β is a power of $\alpha^{(p^d-1)/M}$. A general multirate system as shown in Figure 1, with M bands and subsampled by N. In the case of an STFT filterbank, the analysis, resp. synthesis bank consist of the following filters

$$A_m(z^{-1}) = \sum_{n=0}^{N-1} \beta^{nm} z^{n+1-N} \tag{1}$$

$$C_m(z^{-1}) = \sum_{m'=0}^{M-1} \beta^{-mm'} z^{-m'} \tag{2}$$

This scheme is well known for its fast convolution properties, and is called overlap-add scheme. Swapping synthesis and analysis bank leads to its overlap-save counterpart. As shown in [15], this filterbank can implement an exact linear filtering when correctly designed. Therefore, define the subband filters as follows:

$$D_m(z^{-1}) = \sum_{l=0}^{L-1} \beta^{ml} d_l(z^{-1}) \tag{3}$$

Considering an input $u(z^{-1}) = z^j$ with $j = 0..N-1$, the analysis bank output of the m-th band

$$x_m(z^{-1}) = \beta^{m(N-1-j)} \tag{4}$$

This input is filtered with $D_m(z^{-N})$ (because of the downsampling with N) and fed into the synthesis bank yielding

$$\begin{aligned} y_m(z^{-1}) &= x_m(z^{-1}) D_m(z^{-N}) C_m(z^{-1}) && (5)\\ &= \sum_{m'=0}^{M-1} \beta^{m(N-1-j)} D_m(z^{-N}) \beta^{-mm'} z^{-m'} && (6)\\ &= \sum_{m'=0}^{M-1} \sum_{l=0}^{L-1} \beta^{m(N-1-j)} \beta^{ml} d_l(z^{-N}) \beta^{-mm'} z^{-m'} && (7) \end{aligned}$$

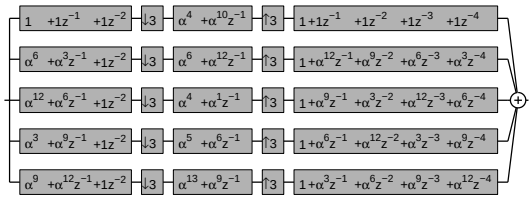

Fig. 2. Filterbank with component codes in each subband for the $\mathcal{R}(15,10,6)$. (example 1)

The filterbank output $y(z^{-1})$ is obtained as the sum over all bands:

$$\begin{aligned} y(z^{-1}) &= \sum_{m=0}^{M-1} y_m(z^{-1}) && (8)\\ &= \sum_{m=0}^{M-1} \sum_{m'=0}^{M-1} \sum_{l=0}^{L-1} \beta^{m(N-1-j)} \beta^{ml} d_l(z^{-N}) \beta^{-mm'} z^{-m'} && (9)\\ &= \sum_{l=0}^{L-1} d_l(z^{-N}) \sum_{m'=0}^{M-1} \sum_{m=0}^{M-1} \beta^{m(N-1-j+l-m')} z^{-m'} && (10) \end{aligned}$$

Looking closely to the double sum, it is seen that the only remaining term is the one with $m' = l+N-1-j$, due to the orthogonality of the roots of unity. Indeed, if $m' \neq l+N-1-j$, the inner summation (over m') equals zero. Therefore,

$$y(z^{-1}) = z^{-N+1+j} \sum_{l=0}^{L-1} z^{-l} d_l(z^{-N}) \tag{11}$$

This last equation only holds when M is chosen large enough, i.e. $M \geq N+L-1$. Before discussing in the next section what happens if this condition is not fulfilled, we will give an example of a filterbank implementing a RS code.

Example 1: Let us construct a filterbank which implements the convolution $y(z^{-1}) = g(z^{-1})u(z^{-1})$ with

$$g(z^{-1}) = \alpha^{10} + \alpha^9 z^{-1} + \alpha^{11} z^{-2} + \alpha^6 z^{-3} + \alpha^9 z^{-4} + z^{-5}$$

This is the generator polynomial of the $\mathcal{R}(15,10,6)$ RS code in $GF(2^4)$ with roots $\{\alpha^k\}_{k=3..7}$. It is used throughout the paper illustrate our techniques. $u(z^{-1})$ and $y(z^{-1})$ represent resp. dataword and (non-systematic) codeword. With $L = N = 3$,

$$\begin{aligned} g(z^{-1}) &= \sum_{l=0}^{2} z^{-l} d_l(z^{-3}) \text{ with} && (12)\\ d_0(z^{-1}) &= \alpha^{10} + \alpha^6 z^{-1} && (13)\\ d_1(z^{-1}) &= \alpha^9 + \alpha^9 z^{-1} && (14)\\ d_2(z^{-1}) &= \alpha^{11} + z^{-1} && (15) \end{aligned}$$

Choosing $M = 5 > L+N-1$ and using Equation 3, the subband filters $D_m(z^{-1})$ are calculated leading to the filterbank shown in Figure 2. The output of this filterbank is a codeword of $\mathcal{R}(15,10,6)$ (at least when the first $N-1$ all-zero samples are ignored, which is assumed from now on). □

III. Critically subsampled filterbanks and RS codes

In this section, it is explained how the linearly time invariant system described above transforms into a periodically time-varying system if the condition $M \geq N + L - 1$ is not met. This is the basic step in understanding the link between some cyclic codes and filterbanks. Imagine L is too large such that $L = M - N + 1 + d$. In that case, a modulo operation should be added to equation 11:

$$y(z^{-1}) = \sum_{l=0}^{L-1} z^{-mod(N-1-j+l,M)} d_l(z^{-N}) \qquad (16)$$

This means that for $j = 0..d-1$, the last $d-j$ coefficients are folded back. Hence, this multirate system has different impulse responses on different time instants j that repeat periodically. Therefore, it is called a periodically time-varying system as opposed to the normal linear time invariant system.

Example 2: Imagine $M = 5$ and $N = 3$ as in the previous example. If $L = 4$, $d = L - M + N - 1 = 1$ and the following impulse responses are obtained:

$$\begin{aligned} u(z^{-1}) &= z^2 y(z^{-1}) = d_0 z^{-0} + d_1 z^{-1} + d_2 z^{-2} + d_3 z^{-3} \\ u(z^{-1}) &= z^1 y(z^{-1}) = d_0 z^{-1} + d_1 z^{-2} + d_2 z^{-3} + d_3 z^{-4} \\ u(z^{-1}) &= z^0 y(z^{-1}) = d_0 z^{-2} + d_1 z^{-3} + d_2 z^{-4} + d_3 \mathbf{z^{-0}} \end{aligned}$$

Note that d_3 is folded back onto z^{-0}. □

For the applications discussed in the next section, it is crucial that the filterbanks are critically subsampled, i.e. the number of bands M equals the subsample factor N. Hence, the condition $M \geq L + N - 1$ is violated. Therefore, critically subsampled filterbanks are not of much interest if subband filtering is aimed for. However, it is shown in this paper that such a periodically time varying critically subsampled filterbanks are exceptionally well suited to implement Reed-Solomon codes and some other cyclic codes.

Theorem 1: Let $\mathcal{R}(\nu, \kappa, \nu - \kappa + 1)$ be a Reed-Solomon code over $GF(p^d)$ of length $\nu = p^d - 1$ and dimension κ. Consider an STFT-based critically subsampled filterbank with M bands ($M|\nu$), subsampled by $N = M$ and with analysis and synthesis bank (respectively $A_m(z^{-1})$ and $C_m(z^{-1})$) defined as in Equation 1. This filterbank will implement the RS code $\mathcal{R}(\nu, \kappa, \nu - \kappa + 1)$ if a root α^i of $\mathcal{R}(\nu, \kappa, \nu - \kappa + 1)$ is assigned to subband m if and only if $i \bmod M = m$. Stated otherwise,

$$D_m(\alpha^i) = 0 \Leftrightarrow \exists j \in \mathbb{Z} | i = Mj + m \qquad (17)$$

proof: Introducing the notation d_{lk} and D_{mk}, with

$$d_l(z^{-1}) = \sum_{k=0}^{K-1} d_{lk} z^{-k} \qquad (18)$$

$$D_m(z^{-1}) = \sum_{k=0}^{K-1} D_{mk} z^{-k} \qquad (19)$$

the following relation holds (assuming $\beta = \alpha^{\nu/M}$, a primitive M-th root of unity):

$$d_{lk} = \sum_{m=0}^{M-1} \alpha^{-\nu m l/M} D_{mk} \qquad (20)$$

The proof of theorem 1 consists in proving that the filterbank output is a codeword of $\mathcal{R}(\nu, \kappa, \nu - \kappa + 1)$ (up to an interleaving), for every input $u(z^{-1}) = z^j$ with $j = 0..N-1$. Therefore, let us recall that the filterbank output for $j = N-1$ equals $y(z^{-1}) = \sum_{l=0}^{L-1} z^{-l} d_l(z^{-N})$. Interleaving this $y(z^{-1})$ gives

$$y^{\Pi}(z^{-1}) = \sum_{l=0}^{L-1} z^{-\nu l/M} d_l(z^{-1}) \qquad (21)$$

Now it is shown that $y^{\Pi}(z^{-1})$ is a codeword of $\mathcal{R}(\nu, \kappa, \nu - \kappa + 1)$ by calculating its Mattson-Solomon polynomial Δ [16]:

$$\Delta(z^{-1}) = \sum_{j=0}^{\nu/M-1} \sum_{m=0}^{M-1} \Delta_{Mj+m} z^{-Mj-m} \text{ with} \qquad (22)$$

$$\begin{aligned} \Delta_{Mj+m} &= \sum_{l=0}^{L-1} \sum_{k=0}^{\nu/M-1} \alpha^{(\nu l/M+k)(Mj+m)} d_{lk} \qquad (23) \\ &= \sum_{l=0}^{L-1} \sum_{k=0}^{\nu/M-1} \alpha^{(\nu l/M+k)(Mj+m)} \sum_{m'=0}^{M-1} \alpha^{-\nu m' l/M} D_{m'k} \\ &= \sum_{k=0}^{\nu/M-1} \alpha^{k(Mj+m)} \sum_{l=0}^{L-1} \sum_{m'=0}^{M-1} \alpha^{\nu l(Mj+m-m')/M} D_{m'k} \end{aligned}$$

This can further be simplified noting that the double sum is only nonzero if $m = m'$, due to the orthogonality of the M-th roots of unity.

$$\Delta_{Mj+m} = \sum_{k=0}^{\nu/M-1} \alpha^{k(Mj+m)} D_{mk} = D_m(\alpha^{Mj+m}) = 0$$

The next step is to show that for all $j = 0 : N-1$, the output of the filterbank belongs to $\mathcal{R}(\nu, \kappa, \nu - \kappa + 1)$. In general, $y(z^{-1})$ is given by equation 16. Interleaving results in

$$\begin{aligned} y^{\Pi}(z^{-1}) &= \sum_{l=0}^{L-1} z^{-mod(\nu(N-1-j+l)/M, \nu M/M)} d_l(z^{-1}) \qquad (24) \\ &= z^{-\nu(N-1-j)/M} \sum_{l=0}^{L-1} z^{-\nu l/M} d_l(z^{-1}) \bmod (1 + z^{-\nu}) \end{aligned}$$

This is a codeword too because it is the original codeword (at $j = N-1$) cyclically shifted by $\nu(N-1-j)/M$, which proves the theorem. ■

Example 3: Continuing our example of the $\mathcal{R}(15, 10, 6)$ code, the roots $\alpha^3, \alpha^4, \alpha^5, \alpha^6, \alpha^7$ are distributed among the polynomials $D_m(z^{-1})$ as follows:

$$\alpha^3, \alpha^6 \Leftarrow D_0(z^{-1}) = \alpha^9 + \alpha^2 z^{-1} + z^{-2} \qquad (25)$$

$$\alpha^4, \alpha^7 \Leftarrow D_1(z^{-1}) = \alpha^{11} + \alpha^3 z^{-1} + z^{-2} \qquad (26)$$

$$\alpha^5 \Leftarrow D_2(z^{-1}) = \alpha^5 + z^{-1} \qquad (27)$$

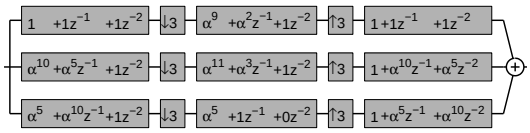

Fig. 3. Filterbank with component codes in each subband for the $\mathcal{R}(15, 10, 6)$. (example 1)

Using equation 20, $d_m(z^{-1})$ is readily calculated:

$$d_0(z^{-1}) = \alpha^1 + \alpha^{13} z^{-1} \quad (28)$$

$$d_1(z^{-1}) = 1 + \alpha^{12} z^{-1} + \alpha^5 z^{-2} \quad (29)$$

$$d_2(z^{-1}) = \alpha^{14} + \alpha^5 z^{-1} + \alpha^{10} z^{-2} \quad (30)$$

Hence, the output of the filterbank ($j = N - 1 = 2$)

$$y(z^{-1}) = \alpha^1 + z^{-1} + \alpha^{14} z^{-2} + \alpha^{13} z^{-3} + \alpha^{12} z^{-4} + \alpha^5 z^{-5} + \alpha^5 z^{-7} + \alpha^{10} z^{-8}. \quad (31)$$

According to the theorem, $y^{\Pi}(z^{-1})$ is a codeword of the $\mathcal{R}(15, 10, 6)$ code.

$$y^{\Pi}(z^{-1}) = \alpha^1 + \alpha^{13} z^{-1} + z^{-5} + \alpha^{12} z^{-6} + \alpha^5 z^{-7} + \alpha^{14} z^{-10} + \alpha^5 z^{-11} + \alpha^{10} z^{-12} \quad (32)$$

This is verified by calculating its Mattson-Solomon polynomial $\Delta(z^{-1})$of which the coefficients of z^{-3} up to z^{-7} are zero:

$$\Delta(z^{-1}) = \alpha^{12} + \alpha^{14} z^{-1} + \alpha^1 z^{-2} + \alpha^4 z^{-8} + \alpha^6 z^{-9} + \alpha^8 z^{-10} + \alpha^3 z^{-11} + \alpha^{14} z^{-12} + \alpha^1 z^{-13} + \alpha^{12} z^{-14} \quad (33)$$

□

Note that if M divides κ, the filterbank implements the quasi-cyclic structure for a RS code [17]. If M and κ are coprime, this quasi-cyclic structure does not exist, however the critically subsampled filterbank does exist. An example is shown in Figure 4 for the $\mathcal{R}(15, 10, 6)$ code with $M = 5$.

Secondly, note that the first subband filter is a non-primitive BCH(5,3,3) code in $GF(2^4)$ with α^3 a primitive 5-th root of unity. It is also cyclic and if ν/M were not prime, the procedure can be applied recursively. The other subband filters are not cyclic. However, a filterbank can be found for them too, but this is out of the scope of this paper. In fact, these results can actually be generalized to the family of BCH codes, see [18] for more details.

Finally, remark that in a critically subsampled filterbank, $x_m(z^{-1})$ and $z_m(z^{-1})$ have κ/M, resp. μ/M coefficients, called subband variables. Therefore, the coefficients of $x_m(z^{-1})$ can be seen as input variables. Hence, a filterbank with the analysis bank replaced by a simple serial to parallel converter implements a RS code too, which is equivalent to the original. In this case

$$A_m(z^{-1}) = z^{m+1-N}. \quad (34)$$

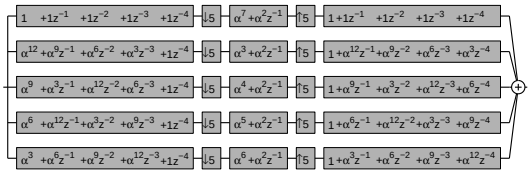

Fig. 4. Filterbank with component codes in each subband for the $\mathcal{R}(15, 10, 6)$. (example 1)

IV. Application: SISO RS decoder

In this section, a SISO decoder for $\mathcal{R}(\nu, \kappa, \nu - \kappa + 1)$ based on the previously developed filterbank decomposition is discussed. First, define the subband samples z_{mk} and x_{mk} as the k-th coefficient of $z_m(z^{-1})$, resp. $x_m(z^{-1})$, similar to Equation 18. Similarly, define a codeword symbol y_{mk} as the $m + kM$-th coefficient of $y(z^{-1})$, i.e.

$$y(z^{-1}) = \sum_{k=0}^{\nu/M-1} z^{-kM} \sum_{m=0}^{M-1} y_{mk} z^{-m}. \quad (35)$$

With a simplified analysis bank (Equation 34), the dataword becomes

$$u(z^{-1}) = \sum_{k=0}^{\kappa/M-1} z^{-kM} \sum_{m=0}^{M-1} u_{mk} z^{-m} \text{ with } u_{mk} = x_{mk}. \quad (36)$$

The soft information of a codeword symbol y_{mk} that is available through the channel output is known as a priori information and is denoted as $\mathbf{P}^a_{y_{mk}}$. Essentially, this is a probability vector (length p^d) indicating the probability that y_{mk} equals the i-th element of the $GF(p^d)$ ($i = 0..p^d - 1$) given the channel output. Any a priori information about the dataword can be incorporated in $\mathbf{P}^a_{x_{mk}}$. Otherwise, it is set to $\mathbf{P}^{equi}$ with equally probable entries.

The SISO RS decoder operates as follows (Algorithm 1): In the initialization step, the extrinsic information of y, z, u is set to $\mathbf{P}^{equi}$. In step 3, the a posterior probabilities (APP) of the subband variables z_{mk} can be calculated since there exists a DFT relation between each consecutive block of z_{mk} and y_{mk}. This operation is performed in parallel by ν/M $SISO_{DFT}$ decoders. In the next step, the (new) extrinsic information about z_{mk} is obtained by a component wise division of its a posterior information by the (old) extrinsic information, followed by a normalization. Note that this step can be skipped in the first iteration. This *extrinsic information* $\mathbf{P}^e_{z_{mk}}$ is then send to 3 SISO decoders corresponding to the 3 subband filters $D_m(z^{-1})$ in step 4. Since the constraint length of these convolutional codes is small, *BCJR's algorithm* in a terminated trellis can be applied to obtain an estimate of the APP of the dataword u_n in conjunction with extrinsic information about the subband variables z_n. Even when there is no a priori information about the dataword, the BCJR decoder comes up with new extrinsic information about z_n since it has redundancy. This extrinsic information is subsequently fed back to the $SISO_{DFT}$ decoders, which can now deliver APP

for the codeword too. After several Gallager-like iterations[1] (step 2), the probabilities (hopefully) converge.

This algorithm is the application of *Gallager's algorithm* to a simple *Tanner graph* with only 3 variable nodes (y_{mk}, z_{mk} and u_{mk}) and 2 parity checks ($SISO_{DFT}$ in between y_{mk} and z_{mk}, $SISO_{BCJR}$ in between z_{mk} and u_{mk}). Note that, according to *Gallager's algorithm*, the SISO decoders only use a priori information about y_{mk} and u_{mk} since there is never extrinsic information originating from any other check than the SISO decoder under consideration.

Algorithm 1 SISO Reed-Solomon decoder

1. Initialization: $\forall k = 0..\nu-1, \forall m = 0..M-1$
   $\mathbf{P}^{e(0)}_{y_{mk}} = \mathbf{P}^{equi}$
   $\mathbf{P}^{e(0)}_{z_{mk}} = \mathbf{P}^{equi}$
   $\mathbf{P}^{e(0)}_{u_{mk}} = \mathbf{P}^{equi}$
2. For $i = 1$ to maximum number of iterations
   3. For $k = 0$ to $\nu/M - 1$
      $([\mathbf{P}^{p(i)}_{y_{mk}}]_m, [\mathbf{P}^{p(i)}_{z_{mk}}]_m) = SISO_{DFT}([\mathbf{P}^{a}_{y_{mk}}]_m, [\mathbf{P}^{e(i-1)}_{z_{mk}}]_m)$
      $[\mathbf{P}^{e(i)}_{z_{mk}}]_m = \text{Normalize}([\mathbf{P}^{p(i)}_{z_{mk}}]_m / [\mathbf{P}^{e(i-1)}_{z_{mk}}]_m)$
      with $[*]_m$ denoting concatenation over $m = 0..M-1$
   4. For $m = 0$ to $M-1$
      $([\mathbf{P}^{p(i)}_{z_{mk}}]_k, [\mathbf{P}^{p(i)}_{x_{mk}}]_k) = SISO_{BCJR}([\mathbf{P}^{e(i)}_{z_{mk}}]_k, [\mathbf{P}^{a}_{x_{mk}}]_k)$
      $[\mathbf{P}^{e(i)}_{z_{mk}}]_k = \text{Normalize}([\mathbf{P}^{p(i)}_{z_{mk}}]_k / [\mathbf{P}^{e(i)}_{z_{mk}}]_k)$
      with $[*]_k$ denoting concatenation over $k = 0..\nu/M-1$

Example 4: Applying algorithm 1 to our example of $\mathcal{R}(15, 10, 6)$ leads to the block diagram in Figure 5. It is based on the filterbank in Figure 3 with a simplified analysis bank (Equation 34). The subband variables z_n are used to exchange information between the codeword y_n and the dataword u_n. In each iteration, using the a priori information (u_n: left; y_n: top) an update of the a posterior probabilities is returned (u_n: right; y_n: bottom). □

Before discussing some simulation results, it is important to mention the equivalence with *Turbo codes*. Turbo codes are in fact parallel concatenated convolutional codes (PCCC) where *BCJR's algorithm* is applied to decode each convolutional code. The component codes are connected by means of an interleaver with the data word, and *Gallager's algorithm* is applied [19]. The proposed decoding scheme for RS codes is very similar. The subbands consist of a number of parallel concatenated convolutional codes, again decoded with *BCJR's algorithm*. These component codes are connected by means of a DFT, as opposed to an interleaver, with the code word. Note that a more complex code -like a DFT- is needed to link the subcodes together in such a way that the overall code has the properties of an MDS code.

Secondly, note the interleaving in Figure 5. The k-th DFT decoder operates on the variables z_{mk} (k fixed) while the the

[1]This principle of exchanging information using several iterations is also called the *Turbo-principle*

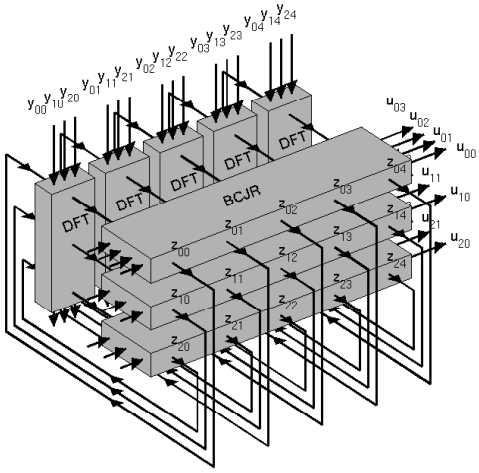

Fig. 5. Block scheme of SISO decoder for $\mathcal{R}(15, 10, 6)$ based on the filterbank shown in Figure 3.

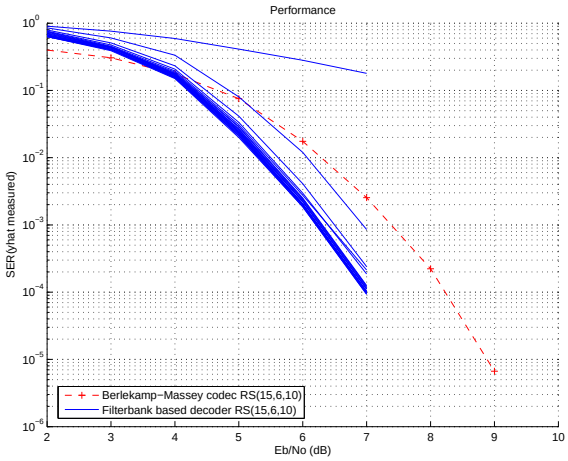

Fig. 6. Simulation results for $\mathcal{R}(15, 6, 10)$

m-th BCJR decoder is connected with z_{mk} (m fixed). Hence, the information is spreaded very fast over the graph.

In Figure 6, simulation results of the $\mathcal{R}(15, 6, 10)$ code are presented. The evolution of the symbol error rate after each iteration of *Gallager's algorithm* is shown by the full lines. In total, 8 iterations are shown. For a symbol error rate of 10^{-4}, and lower, soft decoding using the filterbank based decoder provides a coding gain of approximately 1.5dB compared to a conventional hard decision decoding algorithm, namely *Berlekamp-Massey algorithm*. Remark also that *Gallager's algorithm* rapidly converges after a few iterations.

Simulation results for the $\mathcal{R}(15, 3, 13)$ are summarized in Figure 7. The *Berlekamp-Massey algorithm* as well as the filterbank based decoder are shown. It is interesting to compare this with the third curve which shows a Turbo code, of the same rate 1/5, working in the same Galois field with recursive generator polynomials and a random interleaver. So, this code takes 3 input symbols in $GF(2^4)$, and produces 3 times 5 symbols: the systematic symbols and the convolutions of the

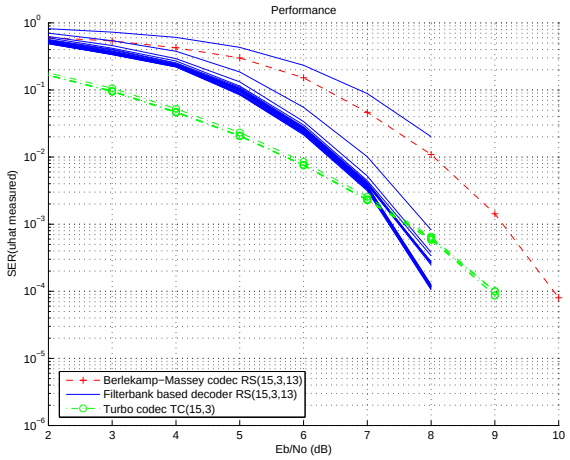

Fig. 7. Simulation results for $\mathcal{R}(15, 3, 13)$ and a Turbo code (rate 1/5, dataword length 3)

3 input symbols with their respective convolutional codes. We are aware of the fact that better generator polynomials may be found, and that Turbo codes in higher order Galois fields and small block lengths do not perform very well. On the other hand, the Turbo (convolutional) component codes are dense and are close to optimal for such a small dataword length. Nevertheless, one can see that the Reed-Solomon codes performs better by almost 1dB (symbol error rate of 10^{-4}) compared to these Turbo codes.

V. Conclusion

This paper focuses on the role of filterbanks in coding applications. Our discussion begins with STFT filterbanks. In most applications this filterbank is explicitly designed to ensure that the filtering operations are linear. However, if the number of bands is not chosen to be large enough, this filterbank acts as a periodically time varying system. Though this is normally considered an undesirable artifact, it is this periodicity that is exploited to build a critically subsampled filterbank for a Reed-Solomon code. This paper proves that a proper distribution of its roots over the subbands is the key element in constructing such a filterbank. The second part of this paper illustrates how such a filterbank is applied to build a SISO decoder for Reed-Solomon codes. It consists of a number of parallel BCJR decoders, each corresponding to a subband code. These are linked together with a number of so-called DFT decoders. The subband variables transfer soft information between the BCJR decoders and the DFT decoders.

Acknowledgment

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.

References

[1] I. Reed and G. Solomon, "Polynomial codes over certain finite fields," *SIAM J.*, vol. 8, no. 2, pp. 300–304, june 1960.

[2] V. Guruswami and M. Sudan, "Improved decoding of reed-solomon and algebraic-geometric codes," *IEEE Trans. Info. Theory*, vol. 45, pp. 1757–1767, 1999.

[3] U. Cheng and G. Huth, "Bounds on the bit error probability of a linear cyclic code over gf(2l) and its extended code," *Information Theory, IEEE Transactions on*, vol. 34, no. 4, pp. 776 – 785, July 1988.

[4] J. Hagenauer, E. Offer, and L. Papke, "Iterative decoding of binary block and convolutional codes," *Information Theory, IEEE Transactions on*, vol. 42, no. 2, pp. 429 – 445, Mar. 1996.

[5] M. Fossorier and S. Lin, "Soft-input soft-output decoding of linear block codes based on ordered statistics," in *Global Telecommunications Conference, 1998. GLOBECOM 98. The Bridge to Global Integration.*, vol. 5, 8-12 Nov 1998, pp. 2828 – 2833.

[6] J. Forney, G., "Generalized minimum distance decoding," *Information Theory, IEEE Transactions on*, vol. 12, no. 2, pp. 125 – 131, April 1966.

[7] R. Koetter and A. Vardy, "Algebraic soft-decision decoding of reed-solomon codes," *Information Theory, IEEE Transactions on*, vol. 49, no. 11, pp. 2809 – 2825, Nov 2003.

[8] S. Benedetto, D. Divsalar, and J. Hagenauer, "Guest editorial concatenated coding techniques and iterative decoding: Sailing toward channel capacity," *Selected Areas in Communications, IEEE Journal on*, vol. 16, no. 2, pp. 137 – 139, Feb 1998.

[9] J. Forney, G.D., *Concatenated Codes*. M.I.T. Press, 1966.

[10] L. Bahl, J. Cocke, F. Jelinek, 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.

[11] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: turbo-codes," *Communications, IEEE Transactions on*, vol. 44, no. 10, pp. 1261 –1271, Oct. 1996.

[12] A. Vardy and Y. Be'ery, "Bit-level soft-decision decoding of reed-solomon codes," *Communications, IEEE Transactions on*, vol. 39, no. 3, pp. 440 –444, Mar. 1991.

[13] A. Scaglione, G. Giannakis, and S. Barbarossa, "Redundant filterbank precoders and equalizers. i. unification and optimal designs," *Signal Processing, IEEE Transactions on*, vol. 47, no. 7, pp. 1988 – 2006, July 1999.

[14] F. Fekri, R. Mersereau, and R. Schafer, "Two-band wavelets and filterbanks over finite fields with connections to error control coding," *Signal Processing, IEEE Transactions on*, vol. 51, no. 12, pp. 3143 – 3151, Dec 2003.

[15] 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.

[16] N. Sloane and F. MacWilliams, *The Theory of Error Correcting Codes*. North-Holland, Amsterdam, 1977.

[17] 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.

[18] G. Van Meerbergen, M. Moonen, and H. De Man, "Filterbank Decompositions for BCH-codes with Applications to Soft Decoding and Code Division Multiple Acces systems," in *Submitted to IEEE International Symposium on Information Theory*, Sept. 4-9 2005.

[19] R. McEliece, D. MacKay, and J.-F. Cheng, "Turbo decoding as an instance of Pearl's Belief Propagation algorithm," *Selected Areas in Communications, IEEE Journal on*, vol. 16, no. 2, pp. 140 –152, Feb. 1998.