

KATHOLIEKE UNIVERSITEIT LEUVEN
FACULTEIT INGENIEURSWETENSCHAPPEN
DEPARTEMENT ELEKTROTECHNIEK
Kasteelpark Arenberg 10, 3001 Leuven (Heverlee)

DESIGN AND APPLICATIONS OF FILTER BANK STRUCTURES IMPLEMENTING REED-SOLOMON CODES

Promotoren:
Prof. dr. ir. M. Moonen
Prof. dr. ir. H. De Man

Proefschrift voorgedragen tot
het behalen van het doctoraat
in de ingenieurswetenschappen

door

Geert VAN MEERBERGEN

December 2006



KATHOLIEKE UNIVERSITEIT LEUVEN
FACULTEIT INGENIEURSWETENSCHAPPEN
DEPARTEMENT ELEKTROTECHNIEK
Kasteelpark Arenberg 10, 3001 Leuven (Heverlee)

DESIGN AND APPLICATIONS OF FILTER BANK STRUCTURES IMPLEMENTING REED-SOLOMON CODES

Jury:
Prof. dr. ir. Y. Willems, voorzitter
Prof. dr. ir. M. Moonen, promotor
Prof. dr. ir. H. De Man, promotor
Prof. dr. ir. B. Preneel
Prof. dr. ir. J.-P. Linnartz
Prof. dr. ir. L. Vandendorpe
Prof. dr. ir. H. van Tilborg
dr. ir. L. Van der Perre

Proefschrift voorgedragen tot
het behalen van het doctoraat
in de ingenieurswetenschappen

door

Geert VAN MEERBERGEN

U.D.C. 621.391 December 2006

© Katholieke Universiteit Leuven – Faculteit Ingenieurswetenschappen
Arenbergkasteel, B-3001 Heverlee (Belgium)

Alle rechten voorbehouden. Niets uit deze uitgave mag vermenigvuldigd en/of openbaar gemaakt worden door middel van druk, fotocopie, microfilm, elektronisch of op welke andere wijze ook zonder voorafgaande schriftelijke toestemming van de uitgever.

All rights reserved. No part of the publication may be reproduced in any form by print, photoprint, microfilm or any other means without written permission from the publisher.

D/2006/7515/91

ISBN 978-90-5682-757-1

Voorwoord

Iedereen die me min of meer goed kent, weet dat schrijven mijn ding niet is. Het is dan ook tot ieders verbazing, mezelf inbegrepen, dat mijn boekje een volle 160 pagina's telt. En ik kan jullie verzekeren, het schrijfwerk is meer dan eens een nachtmerrie geweest. Wat echter veel verklaart is dat dit boekje het verhaal vertelt over een Reed-Solomon code en filterbank die mekaar ontmoeten. Niet echt romantisch, en als je erover begint na te denken besluit je dat dit boekje eigenlijk een hoop wiskundig geneuzel bevat, waarin ik blijkbaar thuis ben. Merkwaardig genoeg steunt een hele hoop technologie, die we dagdagelijks gebruiken, intensief op al die wiskunde. Misschien is dit ook het moment om een schuchtere poging te wagen om de geïnteresseerde lezer – zoals dat heet – te verwijzen naar de rest van deze thesis. Echter, ik besef ten volle dat het aantal mensen dat nu naar pagina 5 gezapt is statistisch irrelevant is. En voor diegenen die overblijven zal ik m'n best doen om een stukje begrijpbare literatuur te produceren in de vorm van dit voorwoord, ditmaal in het Nederlands en niet-wiskundig.

Een eerste iemand die ik aan jullie wil voorstellen is mijn promotor Marc Moonen. Marc en ik ontmoetten elkaar een aantal jaren geleden, hij als Erasmus coördinator en ik als toenmalig Erasmus student in spe. Ik herinner mij toen de omslachtige goedkeuring van mijn programma, waarbij ik Marc steeds mocht ervaren als mijn bondgenoot in de strijd voor mijn Erasmusjaar. Het werd een jaar waar ik nog steeds op terugkijk met veel goede herinneringen en waarvoor ik Marc waarschijnlijk nooit toepasselijk bedankt heb. Marc, bij deze dus. Ook wanneer ik als beginnend doctoraatsstudent het originele onderwerp (iets met MIMO als ik mij goed herinner) aan mijn laars lapte en begon te experimenteren met vreemde begrippen zoals Reed-Solomon codes in Galois velden – waar de wetten van de "gewone" algebra plots op zijn kop stonden – heb je me steeds gesteund. De thesis werd een samengaan van codetheorie met signaalverwerking, en een heleboel "friday meetings" – die steevast doorgingen op maandag, dinsdag, woensdag of donderdag – werden dan ook gespendeerd aan wat jij "vieze woorden" noemde, en hoe we die door een signaalverwerkersbril konden bekijken, d.w.z. zonder "vieze woorden". Je hebt steeds toegelaten dat ik "out of the box" dacht, en dat is dan ook cruciaal geweest voor het welslagen van

deze thesis. Bedankt hiervoor!

Ook mijn copromotor Hugo De Man heb ik steeds gekend als een open-minded onderzoeker. Beste Hugo, dat heb ik erg gewaardeerd. Je hebt de thesis vanop een grotere afstand gevolgd dan Marc, maar ik wil je zeker bedanken voor je suggesties en aanmoedigingen gedurende die periode. De "proficiat" mailtjes die ik steevast kreeg als een paper aanvaard werd zal ik niet vlug vergeten.

De andere leden van het leescomité, Bart Preneel en Liesbeth Van der Perre wil ik bedanken voor het nalezen van de tekst. In het bijzonder wil ik Bart bedanken om zijn appreciatie voor het onderwerp. Als gerenommeerd cryptographie expert had je waarschijnlijk een stapje voor in het ontcijferen van mijn tekst, maar ik heb jouw feedback vanuit code theoretisch perspectief erg gesmaakt.

Ik wil ook Prof. Luc Vandendorpe, Prof. Henk van Tilborg en Prof. Jean-Paul Linnartz bedanken voor hun onmiddellijke bereidheid deel uit te maken van de jury, ondanks hun drukke agenda. Speciaal hierbij apprecieer ik de aanwezigheid van Henk Van Tilborg. Immers een van zijn papers (met als coauteur Solomon zelf) is het vertrekpunt van dit werk.

Een aanzienlijk deel van mijn tijd spendeerde ik in bureau 01.86 (maar ook daarbuiten) samen met Raphael, TJ, Olivier en Geert[1]. Fantastische collega's, vrienden en partners in crime. Deze vier Dalton brothers zorgden ervoor dat de happy chaos in ons bureau voldoende op peil bleef, zodat ik me er altijd thuisvoelde. Een prijs voor slordigste bureau van Vlaanderen bestaat niet en dat vonden we best wel jammer. Ook een uitgebreide collectie gadgets die over de jaren heen (al dan niet rechtmatig) in ons bureau belandde zal me lange tijd bijblijven. Getuige hiervan ook de vele nieuwe collega's die vlug hun weg vinden naar ons bureau (en vooral dan naar de zetel). Het waren fantastische tijden en hopelijk blijven we, na het scheiden van onze wegen naar verafgelegen oorden (zoals daar zijn Australië en Washington, maar vooral Eindhoven en Delft) contact houden. Het ga je goed!

Ik wil ook de hechte groep "anciens" van de DSP club bedanken, waarvan de meesten het Marc-nest reeds ontgroeid zijn: Simon, Geert en Geert (Valois velden?), Gert, Koen, Hilde, Ann en Koen (onze West-Vlamingen), Toon en Imad (onze baby boomers). Bedankt om mij alle broodnodige "truken van de foor" aan te leren die nuttig zijn om te overleven als doctoraatsstudent. Ilse en Veerle (en hun VISA kaart in het bijzonder) wil ik bedanken om mij uit de problemen te helpen als door die "truken van de foor" een afgelegen conferentiehotel mijn solvabiliteit weer eens in vraag stelde.

Until recently, the use of foreign languages was very limited, apart from some West-Flemish, which we tried to ignore. Recently, Marc's group welcomed a

[1]There is no point in hiding your Dutch language capabilities anymore, DSP II presentations are over now.

lot of newcomers from all over the globe. Many thanks to Deepak, Jan, Kim, Paschalis, Prabin, Romain, Sam, Sylwek, Vincent, Ye. All the best with your PhD and if I can give you a tip: already reserve an auditorium for your defense.

Doctoreren kan niet zonder afwisseling, ontspanning en afleiding. Wat dat betreft ben ik mijn vrienden veel dank verschuldigd. In het bijzonder wil ik daarbij Geert, Anneleen, Pieter, Allison, Steven, Natalie, Steven, Joke, Wim, Jan, en Yvan van harte bedanken: voor de klimuitstapjes, voor alle bergjes – van Huldenberg tot Ventoux – waar toch altijd een verbeten strijd wordt geleverd om als eerste boven te zijn, voor "Operatie Screwdriver" en "Operatie Francorchamps", voor de kerstfeestjes zo rond maart, ... Ook Bart wil ik bedanken voor zijn vriendschap. Jouw verwoede pogingen om mij tijdens mijn studententijd alsnog tot tetris kampioen te promoveren zijn me steeds bijgebleven. Ook mijn ex-kotgenootjes, andere oud-studenten en vrienden van Leuven of (ver) daarbuiten en iedereen die ik hierbij vergeet wil ik bedanken voor de vele leuke momenten door de jaren heen.

Als laatste is het thuisfront aan de beurt. Papa, Mama en Broer, bedankt voor de hulp en steun in de keuzes die ik heb gemaakt om de persoon te worden die ik nu ben. Bedankt voor de mogelijkheden die jullie mij geboden hebben om in Leuven en Toulouse te gaan studeren, voor jullie enorme betrokkenheid en interesse in het verloop van mijn doctoraatsonderzoek. Mama, bedankt voor het altijd vertrouwde plekje in Zoersel. Papa, je bent altijd een voorbeeld geweest van topresearch door keihard te werken, voor mij, voor m'n broer maar ook voor vele andere onderzoekers en studenten binnen Philips en de TU/e. Het gaat jullie goed!

Save the best for last: Wine, zonder je enthousiasme, humor, geduld, doorzettingsvermogen, liefde en de niet-in-een-paar-regels-samenvatbare steun bij een tegenvaller zou dit boekje het licht nooit gezien hebben. Wine, nu zal eindelijk het grote geld binnen gaan rollen; begin maar vast met uitgeven. Trust me, I am a doctor[2]...

Geert,
Oktober 2006

[2]Keep on dreaming...

Abstract

In nowadays communication systems, error correction provides robust data transmission through imperfect (noisy) channels. Error correcting codes are a crucial component in most storage and communication systems – wired or wireless –, e.g. GSM, UMTS, xDSL, CD/DVD. At least as important as the data integrity issue is the recent realization that error correcting codes fundamentally change the trade-offs in system design. High-integrity, low redundancy coding can be applied to increase data rate, or battery life time or by reducing hardware costs, making it possible to enter mass market.

When it comes to the design of error correcting codes and their properties, there are two main theories that play an important role in this work. *Classical coding theory* aims at finding the best code given an available block length. This thesis focuses on the ubiquitous Reed-Solomon codes, one of the major achievements of classical coding theory. Nowadays, their applications span 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. However, RS codes have some drawbacks either; A well know problem, addressed in this thesis, is the lack of an efficient soft decoder.

With the introduction of Turbo codes, soft decoding attracted a lot of attention, leading to another theory, commonly referred to as *modern coding theory*. This theory results from the ambition to approach the ultimate limits of communication (Shannon limit) as close as possible. Codes belonging to this class are concatenated codes, typically with a very long block length. This block length is a crucial issue regarding their performance, but prevents application of these codes in consumer electronics, where severe implementation constraints rule. Reducing this block length is not a good option, since the overall code performance of a concatenated code with a small block length is seriously affected by its particular structure.

In this thesis, a novel solution to this problem is presented. Instead of concatenating several small codes, we start from RS codes (with a maximum distance)

which are subsequently broken into a number of component codes. Then, algorithms of modern coding theory (Gallager's algorithm) can be applied to obtain a soft RS decoder. We refer to this technique as *coordinated concatenation*, since the component codes are carefully designed and concatenated such that the overall code is an RS code (with optimal distance properties given a finite block length). Hence, the thesis can best be situated in between classical and modern code theory, since it combines the optimal distance properties of classical codes with the performance of soft decoding methods originating from modern code theory. The development of a soft RS decoder is the most important contribution from a practical point of view.

The RS code decomposition is a crucial aspect in the soft RS decoder. Therefore, the critically subsampled filter bank structure behind RS codes is unveiled, which is the main contribution from a theoretical point of view. The critical subsampling is crucial, and allows us to separate the subband codes (that add redundancy) from the synthesis bank code (that combines the redundancy). The existence of a critically subsampled filter bank is remarkable noting that it can never perform linear filtering operations. In fact, the cyclic character of the RS code is called upon, which matches with the periodically time varying behavior of the filter bank resulting from the critical subsampling. Furthermore, an extension towards the family of BCH codes is presented.

A second (similar) problem arises when an error correcting code is concatenated with an advanced modulator, e.g. an OFDM modulator which is frequently used to tackle the frequency selective character of a communication channel. Although OFDM with bit-loading is known to achieve capacity (i.e. in concatenation with an infinite length, rate 0 theoretical code), the situation is quite different if a finite length code is applied. In the latter case, the distance properties of the overall code must be carefully assessed and optimized.

In this thesis, a new approach regarding this problem is presented, which is once more based on RS codes and their filter bank representation. It is explained how an OFDM scheme can seamlessly be merged with an RS code, leading to a new scheme referred to as RS-OFDM. More specifically, an RS-OFDM system (re)uses the DFT hidden in the synthesis bank of the filter bank representation of an RS code in the OFDM modulator. The overall optimal scheme so obtained is referred to as RS-OFDM, and shows an RS code that is matched to the OFDM-modulator. Apart from a performance gain, the RS-OFDM scheme shows a reduced PAPR.

Korte Inhoud

In hedendaagse communicatiesystemen zijn geavanceerde foutcorrigerende methodes cruciaal om betrouwbare datatransmissie te realiseren over kanalen met ruis. Fout corrigerende codes zijn dan ook een essentiële component in de meeste opslag en (bedrade of draadloze) communicatie systemen, zoals GSM, UMTS, xDSL, CD/DVD. Minstens even belangrijk als de gegevensintegriteit is de recente vaststelling dat foutcorrigerende codes fundamenteel de tradeoffs die gemaakt worden bij het systeemontwerp veranderen. Zeer betrouwbare codes kunnen toegepast worden om de transmissiecapaciteit te verhogen, of om vermogen te besparen en daarmee de werkingsduur van batterijen te verhogen. Codes kunnen zelfs aangewend worden om hardwarekosten te drukken.

Het ontwerp en de eigenschappen van fout corrigerende codes worden bestudeerd in de codetheorie, die kan deze opgedeeld worden in twee klassen: in de klassieke codetheorie worden codes ontworpen met een maximale performantie (afstandseigenschappen), voor een gegeven bloklengte. Deze thesis concentreert zich op de Reed-Solomon (RS) codes, één van de belangrijkste verwezenlijkingen van klassieke codetheorie. RS codes hebben een nooit gezien gamma van toepassingen, gaande van optische en magnetische opslag (CD, DVD, harde schijven, etc.) over hoge snelheidsmodems (zoals ADSL, xDSL, etc.), mobiele communicatie (met inbegrip van GSM, UMTS, etc.) tot zelfs satellietverbindingen. RS codes worden echter ook geconfronteerd met een aantal problemen; een van de voornaamste problemen is het gebrek aan een efficiënte zogenaamde zachte decoder.

Recent, met turbocodes, die het hoofdstuk van de moderne codetheorie inluiden, stond zachte decodering weer helemaal in de belangstelling. Moderne codetheorie vloeit voort vanuit de ambitie om de uiterste limieten van communicatie (zoals gedicteerd door Shannon) zo dicht mogelijk te benaderen. Codes die tot deze klasse behoren zijn geconcateneerde codes, typisch met een zeer lange bloklengte. Deze bloklengte is essentieel voor de performantie, maar verhindert toepassing van deze codes in consumentenelektronica, waar stringente eisen gesteld worden wat betreft vermogengebruik, kost, latency, etc. Het reduceren van deze bloklengte is niet altijd een goede optie, aangezien de

codeperformantie dan sterk afhankelijk wordt van de specifieke structuur van de code, met performantieverlies tot gevolg.

In deze thesis wordt een andere benadering voorgesteld waarbij een RS code wordt opgebroken in een aantal kleinere component codes. Vervolgens kunnen algoritmen van de moderne codetheorie toegepast worden om een zachte RS decoder te construeren. Deze techniek wordt gecoördineerde concatenatie genoemd, verwijzend naar de componentcodes die zorgvuldig ontworpen zijn zodat het geheel nog steeds een grote RS code vormt, met optimale afstandseigenschappen voor een welbepaalde eindige bloklengte. De thesis kan daarom ook best gesitueerd worden tussen klassieke en moderne codetheorie, aangezien de optimale afstandseigenschappen van klassieke codes gecombineerd worden met de performantie van zachte decodeermethodes uit de moderne codetheorie.

Daarom wordt in het eerste deel van deze thesis de structuur van RS codes grondig bestudeerd. Meer specifiek wordt een nieuwe kritisch gedecimeerde filterbankstructuur blootgelegd achter de RS codes. De kritische decimatie is essentieel, en laat toe om de subband codes die redundantie toevoegen te scheiden van de synthesebank. Het bestaan van een kritisch gedecimeerde filterbank is opmerkelijk gezien het feit dat zulk een filterbank geen lineaire filteroperatie kan uitvoeren. De facto wordt er een beroep gedaan op het cyclische karakter van de RS code, dat in overeenstemming kan gebracht worden met het periodiek tijd variërende gedrag van de filterbank (voortvloeiend uit de kritische decimatie). Vervolgens wordt een uitbreiding van de filterbankstructuur voorgesteld naar de familie van de BCH codes.

Een tweede (gelijkaardig) probleem doet zich voor wanneer een foutcorrigerende code geconcateneerd wordt met een (geavanceerde) modulator, zoals een OFDM modulator. Deze laatste wordt vaak aangewend indien het kanaal een frequentie selectief karakter vertoont. Hoewel het algemeen bekend is dat OFDM met bit-loading de theoretisch voorgestelde kanaalcapaciteit behaalt (d.w.z. met theoretische code van rate 0 en een oneindig bloklengte), is de situatie verschillend indien een code met eindige bloklengte wordt gebruikt. In dit geval moeten de afstandseigenschappen van de algemene code zorgvuldig worden afgewogen en geoptimaliseerd.

Deze thesis stelt voor dit probleem een oplossing voor, die opnieuw gebaseerd is op de filterbankrepresentatie van RS codes; het laatste hoofdstuk van deze thesis demonstreert hoe een OFDM-modulatie naadloos kan geïntegreerd worden met een RS code tot een nieuw modulatieschema RS-OFDM genaamd. Meer specifiek wordt de DFT die (in de filterbank representatie) deel uitmaakt van de RS code hergebruikt door de OFDM modulator. RS-OFDM vertoont dan ook een RS code die a.h.w. aangepast is aan de OFDM modulator. Buiten de hiermee gepaard gaande performantiewinst heeft dit systeem nog andere voordelen zoals een lage PAPR (Peak-To-Average Power Ratio).

Glossary

Mathematical Notation

$\mathbf{a}$	vector $\mathbf{a}$
$\mathbf{a}(z^{-1})$	vector $\mathbf{a}$, function of the z–transform variable
$\mathbf{A}$	matrix $\mathbf{A}$
$\mathbf{A}(z^{-1})$	matrix $\mathbf{A}$, function of the z–transform variable
$\mathbf{A}^T$	transpose of matrix $\mathbf{A}$
$\mathbf{A}^H$	Hermitian transpose of matrix $\mathbf{A}$
$\mathbf{A}^{-1}$	inverse of matrix $\mathbf{A}$
$\mathrm{Diag}\,(\mathbf{a})$	square diagonal matrix with vector $\mathbf{a}$ as diagonal
$\mathrm{Diag}\,(\mathbf{A})$	vector with diagonal elements of a matrix $\mathbf{A}$
$a\,[m]$	m–th element of vector $\mathbf{a}$
$\mathbf{a}\,[\mathbf{m}]$	vector obtained by selecting elements indexed by $\mathbf{m}$
$A\,[m,n]$	element on the m–th row and n–th column of matrix $\mathbf{A}$
$\mathbf{A}\,[\mathbf{m},\mathbf{n}]$	submatrix obtained by selecting rows and columns indexed by $\mathbf{m}$, $\mathbf{n}$ respectively.
$a^{\{m\}}$	m–th coordinate of $a \in \mathbb{F}_{q^n}$ over $\mathbb{F}_q$.
$\mathbf{A} \otimes \mathbf{B}$	Kronecker product of matrix $\mathbf{A}$ and $\mathbf{B}$
$\begin{bmatrix} a & b \end{bmatrix}$	row-wise concatenation of elements, vectors of matrices
$\lfloor x \rfloor$	largest integer smaller or equal to $x \in \mathbb{R}$
$\lceil x \rceil$	smallest integer larger or equal to $x \in \mathbb{R}$
$\mathrm{rnd}(x)$	round $x \in \mathbb{R}$ to the nearest integer
$a \vert b$	a divides b
$\vert \cdot \vert$	absolute value
$\vert\vert \cdot \vert\vert_2$	2–norm
$\mathcal{E}\{\cdot\}$	expectation operator
σ_x^2	variance of x
$x \bmod y$	remainder after dividing $x \in \mathbb{N}$ by $y \in \mathbb{N}$
$m : n$	vector of integers between $m \in \mathbb{Z}$ and $n \in \mathbb{Z}$
$m : s : n$	vector of integers between m and n with step size $s \in \mathbb{Z}$
$a \approx b$	a is approximately equal to b
$\mathrm{Null}\,\mathbf{A}$	Nullspace of a matrix $\mathbf{A}$

$\mathrm{Tr}_n(a)$	trace of $a \in \mathbb{F}_{q^n}$ from $\mathbb{F}_{q^n}$ to $\mathbb{F}_q = \mathbb{F}_{p^m}$
$\mathrm{sgn}(x)$	signum function
$A \cup B$	union of two sets A and B
$\mathrm{rank}(A)$	rank of a matrix $\mathbf{A}$
$\mathrm{w}_H(\mathbf{x})$	Hamming weight of a vector $\mathbf{x}$
$\mathrm{d}_H(\mathbf{x}, \mathbf{y})$	Hamming distance between vectors $\mathbf{x}$ and $\mathbf{y}$
$\Pr(X)$	probability of occurrence of event X
$a_k^{[\nu']}(z^{-1})$	k-th polyphase component of $a(z^{-1})$
$\mathcal{L}_1 \square \mathcal{L}_2$	code obtained by a concatenation of an outer code $\mathcal{L}_1$ with an inner code $\mathcal{L}_2$.
$\mathcal{L}_1 \oplus \mathcal{L}_2$	direct sum code

Fixed Symbols

K	filter length of the subband filters
L	total filter length
M	number of subbands, DFT size
N	subsampling factor
z	z–domain variable
$\mathbf{R}_{\mathbf{xx}} = \mathcal{E}\mathbf{x}\mathbf{x}^H$	autocorrelation matrix corresponding to vector $\mathbf{x}$
$\mathbf{0}_N$	$N \times N$ zero matrix
$\mathbf{0}_{M \times N}$	$M \times N$ zero matrix
$\mathbf{J}$	exchange matrix with ones along the main anti–diagonal and zeros elsewhere
$\mathbf{F}_M$	$M \times M$ DFT matrix
$\mathbf{I}_M$	$N \times N$ identity matrix
$\mathbf{\Pi}_M$	interleaver matrix
$\mathbf{G}_{\mathcal{L}}$	generator matrix of a code $\mathcal{L}[\nu, \kappa]$
$\mathbf{G}_{\mathcal{L}}^{V}$	generator matrix of a code equivalent to $\mathcal{L}[\nu, \kappa]$
$\mathbf{G}_{\mathcal{L}}^{sys}$	generator matrix of a systematic code equivalent to $\mathcal{L}[\nu, \kappa]$
$\mathbb{N}$	the set of natural numbers
$\mathbb{N}_0$	the set of natural numbers without 0
$\mathbb{Z}$	the set of integer numbers
$\mathbb{R}$	the field of real numbers
$\mathbb{C}$	the field of complex numbers
$\mathbb{F}$	field
$\mathbb{F}_q$	finite field (Galois Field) of order q
$I(X;Y)$	mutual information between two stochastic processes X and Y
C	Channel capacity
α	ν-th root of unity in $\mathbb{F}_q$
β,γ	primitive elements in $\mathbb{F}_q$
q	order of $\mathbb{F}_q$
$\mathcal{C}(\nu, M, \delta)$	code of length ν (possibly nonlinear)

$\mathcal{L}[\nu,\kappa]$	linear code of dimension κ and length ν
$\mathcal{I}[\nu,\kappa]$	cyclic code or ideal
$\mathcal{M}[\nu,\kappa]$	minimal ideal
$\mathcal{R}[\nu,\kappa]$	Reed-Solomon code
$\mathcal{B}[\nu,\kappa]$	BCH code
$\mathscr{C}_{p,q-1}^{(s)}$	cyclotomic coset over $\mathbb{F}_p$ modulo $q-1$ containing an integer s
$\mathscr{M}^{(k)}(x)$	minimal polynomial of β^k
$\theta^{(s)}(x)$	primitive idempotent
$Z_{\mathcal{I}}$	zeros of a cyclic code $\mathcal{I}[\nu,\kappa]$
$\overline{Z}_{\mathcal{I}}$	non-zeros of a cyclic code $\mathcal{I}[\nu,\kappa]$
$\mathscr{P}(x), \mathscr{P}'(x)$	primitive polynomial
$g_{\mathcal{I}}(x)$	generator polynomial of a cyclic code $\mathcal{I}[\nu,\kappa]$
$h_{\mathcal{I}}(x)$	parity check polynomial of a cyclic code $\mathcal{I}[\nu,\kappa]$
$i_{\mathcal{I}}(x)$	idempotent of a cyclic code $\mathcal{I}[\nu,\kappa]$
n	multiplicative order of q modulo ν
ν	code length
κ	code dimension
$\nu' = \frac{\nu}{M}$	ratio code length and the number of bands
$\kappa' = \frac{\kappa}{M}$	ratio code dimension and the number of bands
$\rho = \frac{\nu}{\kappa}$	rate of a code
δ	distance of a code
λ	basis of $\mathbb{F}_{q^n}$ over $\mathbb{F}_q \subset \mathbb{F}_{q^n}$
$\overline{\lambda}$	complementary basis of λ
$\mathcal{N}$	normal probability density function
$\mathcal{CN}$	normal cumulative probability density function
Pr_w	word error rate
Pr_s	symbol error rate
δ_{ij}	Kronecker delta
Φ	isomorphism
ϕ	phase of downsamplers
$\mathbf{P}$	reliability matrix

Acronyms and Abbreviations

@	at
ADSL	Asymmetric Digital Subscriber Line
APP	A Posteriori Probability
AWGN	Additive White Gaussian Noise
BCH	Bose-Chaudhuri-Hochquenghem
BCJR	Bahl-Cocke-Jelinek-Raviv
BEC	Binary ERASURE Channel

BP	Belief Propagation
BPSK	Binary Phase Shift Keying
BSC	Binary Symmetric Channel
CD	Compact Disk
CDMA	Code Division Multiple Access
CP	Cyclic Prefix
cf.	`confer`: compare with
DFE	Decision Feedback Equalizer
DFT	Discrete Fourier Transform
DSP	Digital Signal Processor
DVD	Digital Versatile Disc
ECC	Error Correcting Code
e.g.	`exempli gratia`: for example
Eq.	equation
EXIT	Extrinsic Information Transfer (Chart)
FDMA	Frequency Division Multiple Access
FFT	Fast Fourier Transform
FIR	Finite Impulse Response filter
GF	Galois Field
GF-OFDM	Galois field OFDM
GMD	Generalized Minimum Distance (decoding)
IDFT	Inverse Discrete Fourier Transform
IMT	International Mobile Telecommunications
ISI	Inter Symbol Interference
i.e.	`id est`: that is
iff	if and only if
IFFT	Inverse Fast Fourier Transform
IIR	Infinite Impulse Response filter
JTIDS	Joint Tactical Information Distribution System
LDPC	Low Density Parity Check
MAP	Maximum a Posteriori
MC-CDMA	Multi-Carrier CDMA
MC-DS-CDMA	Multi-Carrier Direct-Sequence CDMA
MDS	Maximum Distance Separable
MIMO	Multi–Input Multi–Output system
ML	Maximum Likelihood
MLSD	Maximum Likelihood Sequence Detector
MMSE	Minimum Mean Square Error
MUI	Multi-User Interference
OFDM	Orthogonal Frequency-Division Multiplexing
OFDMA	Orthogonal Frequency-Division Multiple Access
P/S	Parallel–to–Serial converter
PAM	Pulse-Amplitude Modulation
PAPR	Peak to Average Power Ratio
PCCC	Parallel Concatenated Convolutional Code

PSTN	Public Switched Telephone Network
QAM	Quadrature Amplitude Modulation
RS	Reed-Solomon
RS-OFDM	Reed-Solomon OFDM
S/P	Serial–to–Parallel converter
SC-CP	Single Carrier with Cyclic Prefix
SISO	Soft–In Soft–Out
SIHO	Soft–In Hard–Out
SNR	Signal–to–Noise Ratio
SPB	Sphere Packing Bound
STFT	Short Time Fourier Transform
TDMA	Time Division Multiple Access
THP	Tomlinson Harashima Precoder
UMTS	Universal Mobile Telecommunications System
VDSL	Very-high-bit-rate Digital Subscriber Line
vs.	versus
WMF	Whitening Matched Filter
w.r.t.	with respect to
ZF	Zero Forcing

Contents

Voorwoord **i**

Abstract **v**

Korte Inhoud **vii**

Glossary **ix**

Contents **xv**

Samenvatting **xix**

1 Introduction **1**

1.1 Context 2

1.1.1 Information Theory: Foundation of Coding Theory . . . 2

1.1.2 Coding Theory 4

1.1.3 This Thesis: In Between Classical & Modern Coding Theory 9

1.2 Problem Statement 9

1.3 Thesis objectives 10

1.4 Thesis Survey and Contributions 11

2 Finite Field and Coding Theory **15**

2.1 Finite Field Theory 16
2.1.1 Finite Field Transforms 18
2.2 Coding Theory 19
2.2.1 (Non-)Linear Block Codes 19
2.2.2 Cyclic Codes 22
2.2.3 Cyclic codes in the Frequency Domain 23
2.2.4 BCH and RS Codes 25

I Reed-Solomon Codes and Filter Banks

3 Reed-Solomon Codes and Filter Banks 31
3.1 Introduction 31
3.2 Cook-Toom's Algorithm: the Basis of a Filter Bank Implementation 32
3.3 Filter Bank with Subcodes in Subbands 37
3.4 Critically Subsampled Filter Bank 40
3.4.1 Step 1: Equivalent RS Code generated by $\mathbf{G}_{\mathcal{R}}^{V}$ 41
3.4.2 Step 2: Equivalent RS Code generated by $\mathbf{G}_{\mathcal{R}^{\Pi_{\nu'}}}^{V}$ 43
3.4.3 Step 3: Equivalent RS Code generated by $\mathbf{G}_{\mathcal{R}^{\Pi_{\nu'}}}^{P}$ 46
3.5 Critically Subsampled IIR Filter Bank for Systematic RS? . . . 47
3.6 Conclusion 49

4 Reed-Solomon Codes and Filter Banks: An Algebraic Approach 51
4.1 Introduction 51
4.2 Reverse Cook-Toom Decomposition 52
4.3 Critically Subsampled Filter Bank implementing RS Codes . . 53
4.3.1 Algebraic Structure of RS Generator Matrix 53

4.3.2 Critically Subsampled Filter Bank based on Matrix Decompositions 57

4.3.3 Circulant Block Structure of $\mathcal{R}^{\Pi_{\nu'}}[\nu, \kappa]$ 60

4.4 Filter Bank Representations for BCH Codes 62

4.4.1 Construction of the Filter Bank in an Extension Field . 62

4.4.2 Transforming the Filter Bank from $\mathbb{F}_{q^n}$ to $\mathbb{F}_q$ 64

4.5 Conclusion 70

5 2-D Cyclic Codes and Filter Banks 71

5.1 Introduction 71

5.2 Minimal Ideal and Idempotent 72

5.3 2-D Cyclic Codes 74

5.3.1 Type I Filter Bank Decomposition of RS Codes 76

5.3.2 Type II Filter Bank Decomposition of RS Codes 77

5.4 Applications 84

5.4.1 Recursive Filter Banks 84

5.4.2 BCH Codes 84

5.4.3 Unequal Error Protecting Codes 88

5.5 Conclusion 91

II Applications

6 Soft-In Soft-Out Reed-Solomon Decoding 95

6.1 Introduction 95

6.2 Optimal Decoding Criteria 97

6.3 MAP Estimate and Gallager's Algorithm 103

6.4 Gallager's Algorithm applied to the Filter Bank Decomposition 107

6.5 Further Research 115

6.6 Conclusion 116

7 Reed-Solomon Codes and OFDM 119

7.1 Introduction 119

7.2 OFDM System Model 122

7.3 Galois Field OFDM 124

7.3.1 Integer Channel (PAM-constellations) 125

7.3.2 Complex Field Channel (QAM constellation) 127

7.4 Combining RS and OFDM:RS-OFDM 129

7.4.1 RS-OFDM with a Serial Zero Forcing Equalizer Design 131

7.4.2 Block ZF and MMSE Equalizer Design 136

7.5 Further Research: OFDMA 141

7.6 Conclusion 144

8 Conclusions 147

Bibliography 151

Publications 159

Curriculum Vitae 163

Samenvatting

Ontwerp en Toepassingen van Filterbankstructuren voor Reed-Solomon Codes

Hoofdstuk 1 : Inleiding

In het digitale tijdperk worden wij constant herinnerd aan het belang van snelle maar ook betrouwbare transmissie en opslag van digitale gegevens. Vermits machines fouten maken worden geavanceerde technieken ontwikkeld om informatie foutbestending te maken. Fout corrigerende codes, die het belangrijkste onderwerp uitmaken van deze thesis, zijn de kandidaat bij uitstek om deze taak te verwezenlijken.

Het verbeteren van fout corrigerende codes is altijd een belangrijk onderzoekstopic geweest. In veiligheidskritische toepassingen is de correctheid van de data zelfs cruciaal. Minstens zo belangrijk als gegevensintegriteit is het recente inzicht dat fout corrigerende codes fundamenteel de tradeoffs veranderen in het systeemontwerp. Bijvoorbeeld in GSM of UMTS kan zeer betrouwbare, lage rate codering toegepast worden om het transmissiedebiet te verhogen (over een band beperkt kanaal), of de levensduur van de batterijen te verhogen (door het transmissievermogen te verminderen) of om hardwarekosten te drukken. Vanuit dit standpunt bekeken bieden fout corrigerende codes nieuwe, unieke uitdagingen.

Fout corrigerende codes voegen steeds *redundantie* toe. De grondlegger van moderne communicatie, Shannon, was er zich van bewust dat redundantie op zich niet volstaat, tenzij de codesymbolen afhankelijk zijn van vele symbolen

van het datawoord, wat we *geheugen* noemen. Geheugen laat toe een hoog transmissiedebiet te realiseren, terwijl de betrouwbaarheid gehandhaafd blijft. Informatietheorie, voortvloeiend uit Shannon's werk, preciseert hoe klein deze redundantie kan zijn en hoe groot het geheugen moet zijn. In sectie 1.1.1 wordt het potentieel van codetheorie kort onderzocht a.h.v. enkele informatietheoretische bounds. Een belangrijke conclusie hieromtrent is dat de foutwaarschijnlijkheid algebraïsch vermindert i.f.v. de decodeercomplexiteit:

$$\Pr_w \approx C^{-\frac{E(\rho)}{\rho}} \tag{0.1}$$

Het ontwerp van goed codes lijkt op het eerste zicht onbelangrijk, omdat Shannon aantoont dat de kanaalcapaciteit kan benaderd worden met om het even dewelke (willekeurige) *lange* code. Het probleem is eerder om goede codes te vinden met *een beperkte lengte en complexiteit* zodat deze geschikt zijn voor implementatie in bv. consumenten elektronica.

Dit doel wordt nagestreefd door de zogenaamde klassieke codetheorie. Klassieke codetheorie, gebaseerd op de twee pijlers van Shannon, beoogt het vinden van de *beste* code, d.w.z. een code die een maximaal foutcorrigerende capaciteit combineert met een minimum aan redundantie en geheugen. De meest prominente bijdrage van de klassieke codetheorie is zonder enige twijfel de uitvinding van de Reed-Solomon codes in 1960 [57]. In dit paper beschrijven Irving Reed en Gustave Solomon hun code die een van de bouwstenen zou worden in vele moderne toepassingen, gaande van optische opslag (CD, DVD) over draadloze en bedrade communicatie (GSM, xDSL) tot zelfs communicatie met ruimtesondes. RS codes hebben optimale afstandseigenschappen, wat zich vertaalt in een optimale foutcorrigerende capaciteit. Echter, het succes van RS codes kan niet alleen toegeschreven worden aan hun wiskundige eigenschappen. Ook de exponentiële groei aan digitale signaalverwerking heeft een belangrijke rol gespeeld. Een andere sleutel tot het succes van RS codes is de ontwikkeling van efficiënte decodeer algoritmen. Klassiek wordt een hard decodeeralgoritme gebruikt: het Berlekamp-Massey algoritme. Desalniettemin zijn er ook enkele nadelen verbonden aan RS codes, zoals we hierna zullen bespreken.

Recent is er veel belangstelling voor de moderne codetheorie. In de moderne code theorie tracht men zeer lange codes te ontwerpen die de uiterste grenzen (Shannon) benaderen. Alhoewel sommige toepassingen een code performantie vereisen die zeer dicht aanleunt bij deze grenzen, vindt moderne code theorie zijn oorsprong eerder in informatietheoretische kringen. De twee pijlers van moderne codetheorie zijn concatenatie en zachte decodering. Concatenatie werd ontwikkeld door Forney [27]. Het principe van concatenatie is het gebruik van een kleine zogenaamde *inner code* die een bescheiden foutenwaarschijnlijkheid realiseert, b.v. $\Pr_w \leq 10^{-2}$, en die bij voorkeur zacht gedecodeerd kan worden. Als *outer code* wordt een lange, hoge rate algebraïsche niet-binaire code aangewend, bijv. een RS code, met een krachtig decodeeralgoritme om de

foutenwaarschijnlijkheid drastisch te verlagen, b.v. $\Pr_w \leq 10^{-10}$. Dit schema toont een exponentieel dalende foutenwaarschijnlijkheid

$$\Pr_w \approx e^{-\nu E_c(\rho)},$$

met een decodeercomplexiteit die slechts algebraïsch is in de code lengte ν (gedomineerd door Berlekamp-Massey's algoritme). Slechts onlangs met het verschijnen van Turbocodes [7] erkent men dat de uitwisseling van zachte informatie tussen de componentencodes de codeperformantie aanzienlijk verbetert. Hoewel dit fenomeen reeds beschreven werd door Gallager in 1963, wordt zachte decodering beschouwd als de tweede belangrijkste bijdrage van moderne codetheorie.

Deze thesis situeert zich dan ook *tussen* de klassieke en moderne codetheorie in. Er wordt verder gebouwd op RS codes, aangezien de combinatie van een minimum aan redundantie en geheugen met een maximum aan performantie noodzakelijk iis. In het bijzonder in consumentenelectronica zijn RS codes een valabele optie, rekening houdend met het verband tussen geheugen (bloklengte) van de code en de uiteindelijk geheugenvereisten op de chip. Desalniettemin moeten principes van moderne codetheorie (zachte decodering en concatenatie) in acht genomen worden. Daarom combineert deze verhandeling RS codes (klassieke codetheorie) met technieken uit de moderne codetheorie.

Welke bijdrage kan moderne code theorie leveren aan RS codes? Het blijkt dat RS codes aan bepaalde nadelen lijden, die zich hoofdzakelijk bij de *interface* met het kanaal afspelen. Bijvoorbeeld in een toepassing waar hogere orde constellaties nutteloos zijn, zou een BPSK symboolfout (bitfout) resulteren in een volledig verkeerd Galois veld (GF) symbool. Typisch, wordt deze moeilijkheid overwonnen door een inner (meestal convolutionele) code te gebruiken, die bij voorkeur zacht wordt gedecodeerd. RS codes opereren dan als outer codes: indien de decoder van de inner code faalt, wat typisch resulteert in een aantal opeenvolgende symboolfouten, wordt de fout corrigerende capaciteit van de RS code benut. De optimaliteit van het gehele (geconcateneerde) systeem wordt echter zelden bestudeerd. In het bijzonder wanneer codes met een beperkte bloklengte gebruikt worden (consumentenelectronica) is dit van belang. Een betere oplossing bestaat erin om 'e'en grote RS code te gebruiken, weliswaar met een zachte decoder die extra informatie kan benutten gerelateerd aan de onzekerheid van elk symbool.

Wat betreft meer geavanceerdere kanalen, zoals frequentie selectieve of meerdere gebruikers kanalen, doen er zich gelijkaardige problemen voor bij het interface tussen de RS code en de modulator. In een typisch systeem wordt het kanaal getransformeerd in een aantal parallelle kanalen door geavanceerde modulatoren zoals OFDM of CDMA. Vervolgens wordt elk van deze parallelle subkanalen beschermd tegen fouten door een outer (bijv. RS) fout corrigerende code. Opnieuw wordt de optimaliteit van het volledige geconcateneerde systeem zelden bestudeerd.

Samenvattend, is de interface met het kanaal naar onze mening de primaire reden die verhindert dat RS codes een nog meer prominente rol aannemen in de huidige communicatie systemen. Zoals hierboven besproken worden deze interface problemen typisch opgelost door een geconcateneerd systeem te ontwerpen. De performantie van het gehele systeem is sterk afhankelijk van de exacte structuur van de componentcodes, in het bijzonder wanneer de bloklengte klein is. De doelstelling van deze thesis is een alternatief systeem voor te stellen, rekening houdend met globale optimaliteit. Hiervoor wordt een omgekeerde benadering voorgesteld: Een RS code wordt opgebroken in een aantal kleinere component codes. Hiervoor wordt een filterbank representatie van een RS code ontwikkeld. Indien deze kleinere codes terug samengesteld worden ontstaat een grote geconcateneerde code, die nog steeds de optimale afstandseigenschappen van een RS code heeft. Dit is een belangrijk principe: aangezien de bloklengte van de componentencodes typisch klein is, is een zorgvuldige aaneenschakeling vereist. Wij verwijzen naar dit type van concatenatie als *gecoördineerde concatenatie*.

De filterbankstructuur die aan de grondslag ligt wordt grondig bestudeerd in deze thesis. Er wordt aangetoond dat er een zeer sterke band is tussen Reed-Solomon codes en filterbanken. De uiteindelijk bekomen filterbankstructuur bestaat voor elke RS code en is kritisch gedecimeerd. Dit is opmerkelijk, aangezien een kritisch gedecimeerde filterbank geen lineaire filtering operaties kan verrichten, die normaal worden gebruikt om een RS code te beschrijven. Niettemin, vanwege de cyclische eigenschappen van een RS code, kan er een verband gelegd worden met filterbanken. Deze structuur laat ons toe de voordien beschreven interface problemen op te lossen. In een eerste toepassing wordt beschreven hoe technieken uit de moderne codetheorie kunnen worden toegepast om een zachte RS decoder te bouwen. Als tweede toepassing worden de RS codes geintegreerd met een OFDM/CDMA modulator. Zodoende kunnen de codes RS per frequentieband/gebruiker samenwerken om 'e'en grote RS code te verkrijgen, optimaal gegeven de eindige bloklengte.

De thesis bestaat uit twee delen en is verder opgebouwd als volgt: Na een inleidend hoofdstuk over eindige veldtheorie (Hoofdstuk 2) wordt de filterbankrepresentatie van RS codes behandeld in Deel I. Het tweede deel, dat meer toepassingsgericht is, maakt gebruik van de filterbankrepresentatie om de interfaceproblemen op te lossen zoals die in de paragrafen beschreven werden.

Hoofdstuk 2 Hoofdstuk 2 bevat achtergrondmateriaal over eindige velden en (klassieke) codetheorie. Alleen de concepten die noodzakelijk zijn voor een verder begrip van deze thesis worden kort voorgesteld. In sectie 2.1 wordt eindige veldtheorie behandeld. Er wordt aangetoond hoe een extensieveld $\mathbb{F}_q = \mathbb{F}_{p^m}$ kan gedefinieerd worden vertrekkende van $\mathbb{F}_p$, en hoe basistransformaties kunnen worden bepaald gebruiken makende van de *trace* operatie. Op deze laatste wordt veelvuldig beroep gedaan doorheen deze thesis. In Sectie 2.2 wordt codetheorie behandeld. Lineaire en cyclische codes worden gedefinieerd. Vooral

deze laatste verdienen extra aandacht omdat de wortels van de generatorveelterm van een cyclische code bepalend zijn voor een Reed-Solomon code. BCH codes worden vervolgens gedefinieerd als speciaal geval van cyclische codes (Sectie 2.2.4). Deze codes hebben interessante afstandseigenschappen, gebaseerd op de keuze van de wortels van de generatorveelterm. RS codes worden dan behandeld als een speciaal geval van BCH codes. Het hoofdstuk wordt beëindigd door een korte bespreking van (systematische) implementaties van de code gebruik makende van lineaire (IIR) filters.

Hoofdstuk 3 In dit hoofdstuk wordt een kritisch gedecimeerde filterbankdecompositie van een code RS voorgesteld. Vanuit een theoretisch perspectief is dit de belangrijkste bijdrage van de thesis. Het uitgangspunt in sectie 3.2 is een klassieke *overlap-add* filterbank, die een lineaire filtering uitvoert. Deze filterbank evolueert geleidelijk naar een kritisch gedecimeerde filterbank in sectie 3.4. Tot slot in sectie 3.5 wordt een kritisch gedecimeerde filterbank voorgesteld die een zogenaamde *systematische* versie van de RS code implementeert. Er wordt aangetoond dat deze laatstgenoemde erg eenvoudig kan verkregen worden als de subband codes in zijn systematische vorm worden herschreven. Deze filterbankrepresentatie laat ons toe om de RS code in subcodes op te breken. Het bestaan van een dergelijke decompositie is opmerkelijk gezien het feit dat dergelijke filterbanken geen lineaire filteroperaties kunnen uitvoeren. De aliasing - die inherent aan de kritische decimatie is gelinked - komt overeen met het cyclische karakter van de RS code. De technieken die in dit hoofdstuk worden voorgesteld zijn algemeen toepasbaar om op het even dewelke RS code $\mathcal{R}[\nu, \kappa]$ mits de bloklengte ν en de dimensie κ niet coprime zijn, zodat het aantal banden M een gemeenschappelijke deler is van ν is κ.

Hoofdstuk 4 Hoofdstuk 4 stelt een andere benadering voor om dergelijke filterbanken te construeren, die ons toelaten om de filterbankdecompositie tot *elke* RS code uit te breiden. De beperking op ν en κ (copriem) wordt verwijderd, zodat er flexibiliteit is in de keus van de coderate. Deze nieuwe benadering die in Sectie 4.3 wordt voorgesteld is zuiver algebraïsch van aard, en is geschikt voor de constructie van filterbanken voor een volledig gamma van RS codes. Sectie 4.4 tot slot stelt een uitbreiding voor naar de meer algemene familie van BCH codes. Men toont dat een BCH code kan geschreven worden als een som van kritisch gedecimeerde filterbanken. Deze bijdrage heeft belangrijke gevolgen vanuit een praktisch standpunt gezien het ons toestaat om de grootte van het veld q en de bloklengte ν onafhankelijk van elkaar te kiezen (in het geval van RS codes is $q = nu - 1$).

Hoofdstuk 5 Het vorige hoofdstuk toont het bestaan van een filterbankstructuur voor BCH codes aan, maar geeft nog weinig inzicht in de specifieke structuren die aan de basis liggen. Daarom wordt in dit hoofdstuk een meer algemeen kader geschetst dat de technieken zoals die worden voorgesteld in het vorig hoofdstuk verklaart. Dit kader wordt gevonden in de context van 2-D cyclische codes. Het blijkt dat de wortels van RS/BCH worden uitgespreid over

de subband codes, die - zoals aangetoond in sectie 5.3 - als BCH kunnen worden gezien. Dit inzicht leidt tot een aantal praktische resultaten in sectie 5.4. Meer specifiek kunnen recursieve filterbanken (sectie 5.4.1) of codes met een ongelijke fout corrigerende capaciteit (sectie 5.4.3) geconstrueerd worden vanuit de vorige theorie.

Hoofdstuk 6 In het tweede deel van deze thesis worden oplossingen aangereikt van de twee voornoemde problemen. Hoofdstuk 6 behandelt een eerste toepassing waarin de filterbankdecompositie gebruikt wordt om een zachte RS decoder te bouwen. Na een inleiding over optimale decodeercriteria (Sectie 6.2), wordt het MAP-criterium geselecteerd. In sectie 6.3 wordt het algoritme van Gallager vervolgens voorgesteld als een suboptimale techniek die toelaat de complexiteit te reduceren. In Sectie 6.4 wordt gezien hoe het algoritme van Gallager wordt aangewend om een zachte RS decoder te maken gebaseerd op de filterbankdecompositie.

De ontwikkeling van een efficiënte SISO RS decoder is een fundamenteel open probleem in codetheorie, die recent in de belangstelling staat. De filterbankrepresentatie die in Deel I wordt ontwikkeld is een belangrijke stap in het vinden van een geschikte decompositie voor RS codes. De filterbankrepresentatie laat toe om 'e'en grote RS code op te breken in een aantal subcodes met een *sparse* interconnect. De subband variabelen die zo bekomen worden spelen een centrale rol in deze decompositie, omdat ze toelaten het SISO decodeer probleem op te breken in kleinere SISO decoders die informatie uitwisselen zoals gedicteerd door Gallager's algoritme. Vanuit een praktisch overwegingen is de ontwikkeling van een SISO RS decoder de belangrijkste bijdrage van deze thesis.

Hoofdstuk 7 In hoofdstuk 7, wordt een tweede toepassing voorgesteld. Na een korte inleiding over OFDM (Sectie 7.2), is het uitgangspunt een OFDM-systeem waarin de circulant kanaalmatrix in het Galois veld i.p.v. in het complexe veld wordt ontbonden. Het aldus verkregen systeem wordt GF-OFDM genaamd. Om met de eindige veldoperaties compatibel te zijn, wordt het kanaal gekwantiseerd verondersteld en wordt een modulo operatie toegvoegd aan de ontvanger. Gebruik makende van de in deel I ontwikkelde filterbankrepresentatie van een RS code wordt in sectie 7.4 uitgelegd hoe een GF-OFDM systeem naadloos kan worden geïntegreerd met een Reed-Solomon code, resulterend in een nieuwe systeem RS-OFDM genaamd. Het co-design van een RS code en OFDM frontend resulteert in een significante performantiewinst (tot 10 dB in sommige scenarios), die door simulaties wordt aangetoond. Naast de performantiewinst heeft het RS-OFDM systeem extra voordelen zoals een lage PAPR en complexiteit.

Chapter 1

Introduction

In the digital age, we are constantly reminded of the importance of fast but also reliable transmission, storage and retrieval of digital data. Machines however never operate flawlessly, making errors just as humans do. Just as bridges designed to withstand earthquakes, sophisticated techniques are developed to protect information against the digital manifestation of Murphy's law. Error correcting coding, which is the main topic of this thesis, is a proven candidate to accomplish this task.

Error correcting codes have always been an important area in communications, due to an increasing importance of reliable data. Certain data cannot just be wrong, let us think for example of airplanes or bank transfers. But also the performance of mass storage devices is seriously affected by errors, thereby increasing the response time and sometimes requiring human intervention. At least as important as the data integrity issue is the recent realization that error correcting codes fundamentally change the trade-offs in system design. Let us illustrate this with an example of mobile personal wireless communication, e.g. GSM or UMTS. High-integrity, low redundancy coding can be applied to increase data rate (over a band limited channel), or battery life time (by reducing transmit power) or by reducing hardware costs, making it possible to enter mass market. A second example can be found in the field of data storage. Again, sophisticated error correcting codes make it possible to increase the capacity as well as the the throughput of mass memory applications, as can nowadays be found in everyone's PC. From this point of view, error correcting coding offers a unique competitive edge to many diverse areas. This ubiquitous character of error correcting codes accounts for the decision not to focus on a particular application or field in this thesis. Nevertheless, we refer to many applications throughout this work.

In this chapter, the context in which this thesis can be situated is given in Section 1.1. The foundation of coding theory is information theory, which is briefly covered in Section 1.1.1. Information theory determines a number of bounds under which reliable communication is infeasible. These theoretical limits of communication illustrate the potential of coding theory. In Section 1.1.2, it is seen that coding theory itself can be split into classical and modern coding theory, depending on the emphasis. Classical coding theory aims at finding the *best* code for a certain block length. Modern coding theory rather stems from an information theoretic basis, namely to approximate Shannon's limit. In Section 1.1.3 finally, it is explained how this thesis can be situated in between classical and modern coding theory: in this thesis, it is investigated how the results that stem from modern coding theory can be applied to *classical* codes. A detailed description of the problems that are solved can be found in Section 1.2, followed by the thesis objectives in Section 1.3. Finally, a chapter-by-chapter overview with the main contributions of the thesis is given in Section 1.4.

1.1 Context

1.1.1 Information Theory: Foundation of Coding Theory

In view of the previous paragraph, it is not very surprising that coding theory has a rich history. Prior to Shannon's milestone paper in 1948 [64], communication engineers recognized that adding redundancy was useful to increase reliability, e.g. by repeating a message several times, aiming to correct it with a majority vote. The code symbols do not carry the amount of information as in the uncoded case, hence the transmission is redundant. *Redundancy* is the first key concept of coding. The problem of such a repetition code is the reduction of the throughput per channel use. The missing concept, introduced by Shannon, was the *memory*. Shannon realized that redundancy itself accomplishes little, unless the code symbols depend on many data word symbols, which we call *memory*. Encoding and decoding with memory could circumvent the throughput penalty while maintaining the reliability. More specifically, Shannon showed that with a fixed (small) proportion of redundancy, high reliability is possible. Shannon's cornerstone paper, and the resulting information theory, exactly answers how small this redundancy and how large the memory should be. Shannon identifies the fundamental issues concerning information theory, and also solved them, to a large extent. Let us start with Shannon's main theorem.

Theorem 1.1 *For every memoryless channel, there exists a parameter C referred to as the channel capacity, such that the probability of error can be made arbitrarily small by using an appropriate code (and decoder) with rate $\rho < C$.*

The channel capacity C (probably the most celebrated quantity in information theory) is the mutual information between the input X and output Y of a channel, maximized over all possible input distributions

$$C = \max_{\Pr(x)} I(X;Y) \text{ bits/channel use}$$

This formula allows to calculate the bounds under which communication is infeasible.

In theory, the capacity can be achieved by an error correcting code with rate ρ, infinite block length and an optimal decoding strategy, typically with infinite complexity.

A logical second and more practical concern is the dependence of code performance on the block length or memory, corresponding to the second principle of coding. As specific good codes are hard to identify (see the next section), Shannon and Gallager cleverly avoided this problem by analyzing the ensemble of all codes with a given set of code parameters. Because at least one code in the ensemble has a performance matching the ensemble average, the existence of good codes is guaranteed as the block length increases. The Fano [23] and Gallager [32, 33] bounds were introduced as efficient tools to determine the code behavior as a function of the block length, providing informative results up to the ultimate capacity limit. More in detail, Gallager's exponential upper bound explicitly shows that the probability of (codeword) error $\Pr_w$ of the ensemble block code of length ν and rate ρ decreases exponentially with the block length:

$$\Pr_w \leq e^{-\nu E(\rho)}\,. \tag{1.1}$$

Typically, the error exponent $E(\rho) > 0$ for all rates $\rho < C$. Assuming a linear relation for $E(\rho) = E(0) - \rho$, a simple upper bound known as the *union-Chernoff bound* or *union-Bhattacharyya bound* is obtained.

An interesting *lower* bound is the Sphere Packing Bound (SPB) due to Shannon [66, 65], shown in Figure 1.1 for various code rates. This figure lays a theoretical foundation for the large block length (or memory) needed by good codes: the Shannon limit for a specific rate ρ is the horizontal asymptote of the SPB. The difference between the SNR for a certain code dimension κ and the Shannon limit, the so-called SNR gap $\Delta\,SNR$, decreases with growing block length. While the rather lengthy equations involved can be found in [20], we limit ourselves to a simple approximation valid for large blocks only:

$$\kappa\left(\Delta\,SNR_{dB}\right)^2 = \left(\frac{40}{ln10}\right)\underbrace{\left(\rho\frac{2^{2\rho}+1}{2^{2\rho}-1}\right)}_{1.55\pm0.11}\Pr{}_{w,dB}^{-1}. \tag{1.2}$$

With the factor slowly varying between 1.44 and 1.66 for rates between 0 and 1, and with an error rate $\Pr_w = 10^{-4}$, the SNR gap can be approximated as

$$\kappa\left(\Delta\,SNR_{dB}\right)^2 \geq 1000\,.$$

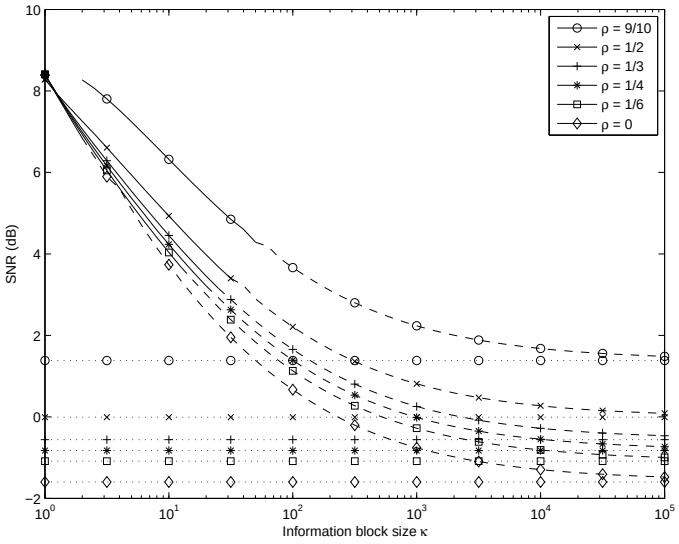


Figure 1.1: Sphere packing bound for various code rates.

This rule of thumb can be used to quickly evaluate the SNR gap, e.g. codes with a data word length of $\kappa = 1000$ can never get closer to capacity than 1 dB.

1.1.2 Coding Theory

In Section 1.1.1, the potential of coding theory is briefly explored by looking into the information theoretical bounds. We did not specify a code that achieves these bounds, our exposition was rather based on the notion of an ensemble of codes. Gallager's exponential error bound (Equation 1.1) decreases exponentially with the block length ν, and therefore long codes are needed. It is clear that to achieve these bounds, an optimal decoding strategy is needed, with a decoding complexity typically exponential in the number of codewords ($\approx O\left(e^{\nu\rho}\right)$) (See also Section 6.2). Using Equation 1.1, the probability of error is seen to decrease only *algebraically* with decoding complexity.

$$\mathrm{Pr}_w \approx C^{-\frac{E(\rho)}{\rho}} \,. \tag{1.3}$$

It is well understood that the key obstacle to practical methods approaching capacity is *not* the design of good codes, any long (random) code will do.[1] Rather, it is a problem of finding good codes *with a moderate length* and still a *reasonable decoding complexity*, suitable for implementation in e.g. consumer

[1] Often, a reference is made to the expression "All codes are good except those that we know of." [82]. i.e. a randomly chosen, long code will turn out to be good with a high probability.

electronics. Therefore, in this section, we take up the constructive aspect of the coding problem.

coding theory is essentially concerned with solving three problems:

- The design of good codes
- The development of decoding algorithms for such codes
- The implementations of these codes in hardware.

Depending on the emphasis, coding theory can be divided into classical or modern coding theory.

Classical Coding Theory & Reed-Solomon Codes

Based on Shannon's two pillars of coding, memory and redundancy, classical coding theory aims at finding the *best* code, i.e. a code that combines a maximum error correcting capacity with a minimal amount of redundancy and memory. The error correcting capacity is determined by a distance measure as explained in Section 6.2, while redundancy and memory refer to the rate and block length respectively. This was extremely important since, in the early days of communication, the allowable (hardware) complexity was very limited. The first "large scale" application of forward error correction is the use of the half-rate Odenwaller convolutional code in satellite communication.

Without any doubt, the most prominent contribution of classical coding theory is the invention of the Reed-Solomon (RS) codes in 1960 [57]. With their paper entitled "Polynomial Codes over Certain Finite Fields", Irving Reed and Gustave Solomon laid the foundations for coding in many modern applications from optical and magnetic storage (CD, DVD, Blu-Ray, Hard Disks) over wireless and wireline communications (GSM, xDSL) to even deep space communications. For example, RS codes make it possible to listen to a scratched CD, to squeeze 52 Megabits per second over the public switched telephone network (PSTN), to pack more than 4.7 GB onto 1 DVD or to send pictures of the outer planets back to earth.

From a mathematical point of view, an RS code $\mathcal{R}[\nu, \kappa]$ is a linear block code defined over a Galois field (GF) (See chapter 2 for an in depth discussion). Linear refers to the κ-dimensional linear subspace (in a ν-dimensional space) that contains all codewords. Intuitively, a codeword $\mathbf{y}$ is obtained from evaluating a data word polynomial $u(x)$ in ν points. These ν values *overdetermine* the data word. Due to specific properties of finite fields, the data word can be recovered from any subset of κ values. Conceptually, the RS code specifies a data word by a curve of ν points. And just as a couple of erroneous points in what is meant to be a smooth curve can be detected (and corrected) by the eye,

the RS code can detect incorrect values and recover the original message. In their paper, Reed and Solomon show that up to t errors can be corrected with $2t \leq \nu - \kappa$. This corresponds to a Hamming distance of at least $\delta = \nu - \kappa + 1$ between each pair of codewords.[2] In chapter 2, it is seen that this is the maximal Hamming distance possible (among all codes with the same dimension κ and length ν). Codes with this property are called *Maximum Distance Separable* (MDS) codes. Hence, RS codes combine a maximal performance with a minimal amount of memory (block length) and a minimum of redundancy. For example, the popular $\mathcal{R}\,[255, 233]$ code can correct up to $t = 11$ byte errors with only 8.5% overhead. This interesting property is (among others) caused by the definition of the RS code in a high order Galois field, and operates on groups of bits – such as bytes – rather than individual zeros and ones. This feature makes RS codes particularly suited for dealing with burst errors. This will be discussed in detail later on.

The success of RS codes cannot only be attributed to their nice mathematical distance properties. Only 20 year after their discovery, RS codes were incorporated in the JTIDS deep space communication standard and in commercial devices such as CD players. This can partly be explained by the considerable growth of digital circuitry which was not available at the time of their invention. Another key in the success of RS codes is the development of efficient decoding algorithms. The most famous is the Berlekamp-Massey hard decoding algorithm, named after their co-inventors Elwyn Berlekamp and Jim Massey [6, 49].

RS codes are therefore an outstanding example of classical code design, from the design of the codes itself over decoder design and hardware implementation. Due to their limited block length in combination with maximal distance and low complexity (albeit hard) decoding algorithms, they are the preferred code in many consumer electronics par excellence. Nevertheless, RS codes face some problems as discussed later.

Modern Coding Theory & Turbo Codes

On the other hand, modern coding theory rather stems from an information theoretic point of view (achieve the Shannon bound) than from practical considerations, although some applications (e.g. deep space communication) require a code performance that approximates the Shannon limit very closely. Only slightly older than its classical counterpart, the cornerstones of modern coding theory are *concatenation* and *soft-decoding*, due to Gallager's book published in 1963 [34].

[2] This fact can easily be verified as follows: if $\delta = 2t + 1$, spheres of radius $t = \left\lfloor \frac{\delta - 1}{2} \right\rfloor$ around each codeword do not overlap. Imagine a vector $\tilde{\mathbf{y}}$ is received which is inside the sphere around $\mathbf{y}$, then it is still closer to $\mathbf{y}$ than to any other codeword. Thus nearest neighbor decoding will correct the error [67].

Let us start with the concept of concatenation, further developed by Forney [27]. Forney developed a class of codes and decoders that have a probability of error decreasing exponentially with block length, while a decoding complexity increasing only algebraically. He arrived at a multilevel coding structure, with an inner and outer code. A short inner code realizes a modest error probability, e.g. $\Pr_w = 10^{-2}$, by (preferably soft-) decoding removing small noise. As outer error correcting code, a long, high-rate algebraic non-binary code (often a RS code) is used with a powerful decoding algorithm to drastically reduce the error rate to industry standards, e.g. $\Pr_w = 10^{-10}$. The burst error correcting capacity is called upon, since mistakes made by the inner decoder have a burst character. This scheme shows an exponentially decreasing error probability

$$\Pr_w \approx e^{-\nu E_c(\rho)},$$

while the decoding complexity is only algebraic in ν (dominated by Berlekamp-Massey's algorithm for decoding the outer (RS) code).

Only recently with the appearance of Turbo codes [7], it was recognized that the exchange of soft-information between the component codes of a concatenated code greatly improves performance. Although already established by Gallager, soft-decoding is regarded as the second main contribution of modern coding theory.

It is instructive to see how modern coding theory dominates professional applications, e.g. deep space communication. This is not very surprising seen the extremely low SNR dealt with. A second reason is that these applications – unlike consumer electronics – are less affected by constraints like power consumption, cost, latency, etc. This creates an ideal environment between research and consumer product development. Comparisons with various concatenated coding systems used in deep space communication can be found in [20] and Figure 1.2. This figure shows code imperfectness as a function of information block size. Imperfectness is the discrepancy between the actual simulated performance of the code and the SPB with the same rate and block size (Equation 1.2). As can be seen, throughout the years, Shannon is approached with steps of half a dB corresponding to successive deep space missions. The Voyager mission uses an outer RS code $\mathcal{R}(255, 223)$ in combination with a small convolutional inner code and interleaver depth of 4. Longer variations of these codes are used for the Cassini and Galileo high gain antenna. When the latter failed to unfurl, a new code was designed to be used by the low gain antenna. This concatenated code was soft, iteratively decoded in four stages, and is probably the first application of soft iterative decoding. Nowadays, a family of Turbo codes (parallel concatenation of convolutional codes) is used with varying rates between $\frac{1}{2}$ and $\frac{1}{6}$ as shown in Figure 1.2. The reason for using a convolutional code instead of an RS code as outer code is the lack of an efficient soft RS decoder: the performance of a concatenated code scheme decreases dramatically if one of the component codes is decoded in a hard fashion.

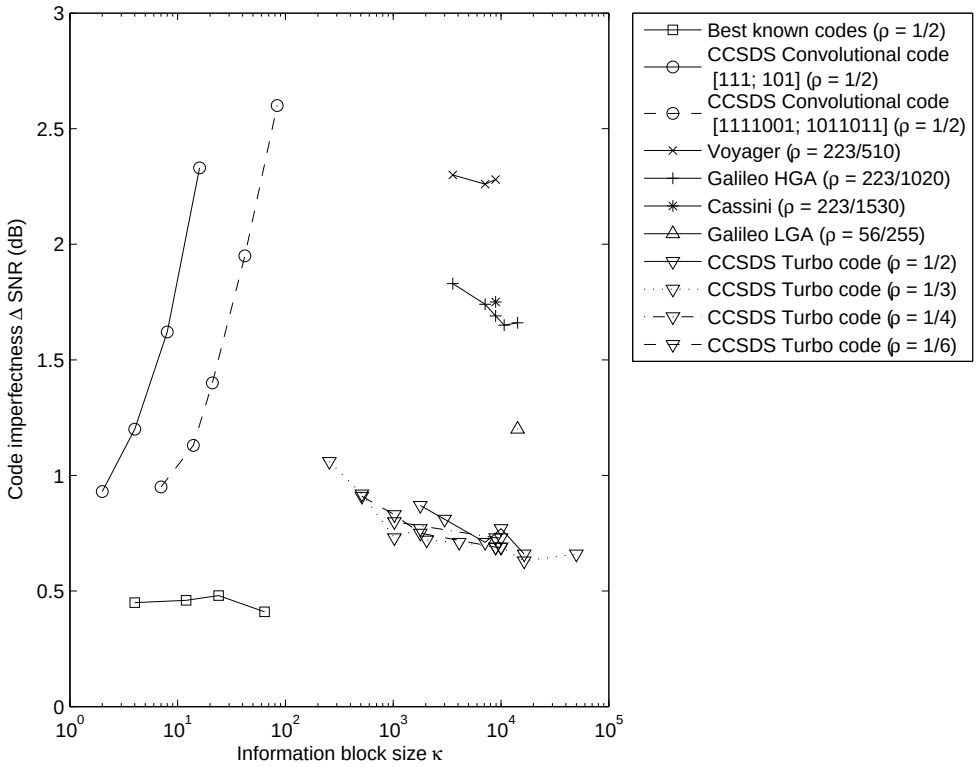


Figure 1.2: Code imperfectness vs. block size for various codes ($\mathrm{Pr}_w = 10^-4$). At the left hand side, a number of minimum distance rate $\frac{1}{2}$ (classical) codes can be found: the $\mathcal{L}[8,4]$ Hamming code, the $\mathcal{L}[24,12]$ Golay code, the $\mathcal{L}[48,24]$ Quadratic residue code and the extended BCH code $\mathcal{B}[128,64]$. At the right hand side, a number of concatenated codes that stem from modern coding theory can be found (as described above). The CCSDS family of turbo codes shown in the lower right corner are based on plain (non-recursive) rate $\frac{1}{2}$ convolutional codes: the CCSDS standard code with connection vectors $[1111001; 1011011]$ and the code with connection vectors [111; 101].

1.1.3 This Thesis: In Between Classical & Modern Coding Theory

This dissertation is built upon RS codes, since the combination of a minimum of redundancy and memory with a maximum of performance is desirable. Keeping in mind that there is a strong relationship between memory (block length) of the codes and memory requirements on the chip, RS codes are definitely worth investigating, especially for consumer electronics. Consumer electronics in general cope with very tight constraints such as cost, power consumption (for wireless), latency and throughput. Therefore, it is not an option to use modern codes like the ones used in deep space communication. Decreasing the block length of these codes is not always a viable alternative, since these codes' strength relies on their block length. On the other hand, classical coding theory taught us how to construct codes with a block length as short as possible. The ubiquitous RS code is a good example, and is the main topic of this dissertation. Nevertheless, the principles of modern coding theory (soft-decoding and concatenation) should be taken into account, and this dissertation can therefore be situated *in between* classical and modern coding theory.

1.2 Problem Statement

What does modern coding theory add to classical codes like RS? It turns out that RS codes suffer from some drawbacks. The problems they face can mainly be situated at the *interface* with the channel. A first problem is related to the higher order Galois fields RS codes operate in. If an application is considered where higher order modulations are useless (e.g. deep space communication), a BPSK bit error would typically result in an entirely wrong GF symbol. This problem is overcome by concatenating an inner (mostly convolutional) code that is preferably soft decoded such that reliability information is taken into account. RS codes operate then as outer codes: if the inner code makes an error, this probably results in a number of consecutive symbol errors that are corrected by the RS code. A better solution would be to use a *soft RS decoder*, which can exploit additional information like the uncertainty about one bit in an entire GF symbol.

Moreover if the application already uses higher order modulations, one would expect RS codes to have an additional advantage of operating in high order Galois fields: after all, increased bit rates in consumer electronics are often attained by increasing the constellation size, e.g. going from ADSL to VDSL. Now, the size of the Galois fields and constellations reach the same order of magnitude. An error of a constellation symbol is then directly translated into a Galois field error, that can be corrected by the RS code. Unfortunately, at the decoder side, the RS decoder must be able to exploit soft-information. I.e. the

correct constellation point (Galois field symbol) can often be found in the *vicinity* of the incorrect constellation point. Therefore, what is actually needed is an efficient *soft RS decoder*. Since this is an open problem, a concatenated system is often called upon, where the stream of constellation points is first converted to bits before entering a decoder as described in the previous paragraph.

Summarizing, the interface with the channel in our opinion is the primary reason that prevent RS codes from taking an even more prominent role in nowadays communication systems. As explained, these problems are typically solved with concatenation (Forney). Concatenation – after RS codes – is a second important keyword in this dissertation. Concatenation is closely interrelated with soft-decoding, to tackle the complexity.

When more advanced channels are considered, such as frequency selective or multiuser channels, similar problems arise at the interface between the RS code and the modulator. In a typical system, the channel is rendered into a number of parallel channels using advanced modulators like OFDM or CDMA. Then, each of these parallel subchannels is protected against errors by a smaller (possibly RS) error correcting code. Again, the optimality of the overall concatenated system of an RS code with an OFDM modulator is rarely considered.

1.3 Thesis objectives

To solve the aforementioned problems, the RS code should be broken into a set of smaller component codes. Hence, a concatenation of the component codes in a strictly defined way generates a large RS code. This is an important principle: since the block length of the component codes is typically small, careful concatenation is required. We refer to this type of concatenation as *coordinated concatenation*.

For this purpose, a well defined structure underlying RS codes should be found. This dissertation shows a very tight connection between RS codes and filter banks. We show that any RS code can be written as a critically subsampled filter bank. This is remarkable, since a critically subsampled filter bank cannot implement a linear filtering operation, which is normally used to describe an RS code. Nevertheless, due to the cyclic properties of an RS code, a close link can be identified with filter banks.

Coordinated concatenation is called upon to solve the interfacing problems described before. A first application describes how techniques originating from modern coding theory can consequently be applied to tackle the complexity of soft RS decoding. As a second application, near the end of this dissertation, it is seen that RS codes can be made compliant with an OFDM/CDMA modulator. Remarkably, component codes per frequency band/user can cooperate to

obtain one large RS code, which is optimal for a given finite block length.

1.4 Thesis Survey and Contributions

The thesis consists of two parts. Following an introductory chapter on finite field theory (Chapter 2), the filter bank representation of RS codes is covered in Part I of this thesis. In the second part, the filter bank representation is applied to solve the interface problems as described in the previous section (See also Figure 1.3).

Chapter 2

Chapter 2 presents some background material on finite fields and (classical) coding theory. Only the concepts that are further used in this thesis are briefly presented. We start by explaining some basic finite field theory (Section 2.1). It is shown how to construct an extension field $\mathbb{F}_q = \mathbb{F}_{p^m}$ starting from a basis in $\mathbb{F}_p$. The finite field counterpart of the well known Discrete Fourier Transform (DFT) is covered. This transform plays an important role in the second section which covers basic coding theory (Section 2.2). This section is started by giving a proper definition of linear and cyclic codes. The roots of the generator polynomial of a cyclic code deserve some additional attention, since they are important in defining an RS code. BCH codes are defined as a special case of cyclic codes (Section 2.2.4). They have interesting distance properties based on the choice of the roots of the generator polynomial. Next, RS codes are covered as a special case of BCH codes. The chapter is ended by a brief discussion on (systematic) encoder implementations using linear (IIR) filters.

Chapter 3

In this chapter, a critically subsampled filter bank decomposition of an RS code is derived. From a theoretical perspective, this is the main contribution of the thesis. The starting point is a classical overlap-add filter bank implementing an RS code in Section 3.2. This filter bank is modified and gradually evolves towards a critically subsampled filter bank in Section 3.4. Finally, in Section 3.5, a critically subsampled filter bank implementing a *systematic* version of the RS code is presented.

These filter bank representations allow us to break the RS code into convolutional subcodes. The existence of such a decomposition is remarkable noting that critically subsampled filter banks can never perform linear filtering operations. It is seen that the aliasing - inherently connected to the critical

subsampling - corresponds with the cyclic character of the RS code. The techniques presented in this chapter can be applied to any RS code subject to the constraint that the block length ν and dimension κ can not be coprime, such that the number of bands M is a common divisor of ν and κ.

Chapter 4

Another approach to construct filter bank representations of RS codes is presented in Chapter 4. This method allows us to extend the filter bank decomposition to *any* RS code without any constraints on ν and κ. As the coprimeness constraint on ν and κ is removed, there is flexibility in the choice of the code rate. This approach presented in Section 4.3 is purely algebraic in nature, since this representation is more suited for building filter banks for an entire gamut of RS codes. Finally, Section 4.4 presents an extension towards the more general family of BCH codes. It is shown that a BCH code can be represented as a sum of critically subsampled filter banks. This contribution has important consequences from a practical point of view since it allows us to choose the field size q and block length ν independently (in the case of RS codes, $q = \nu - 1$).

Chapter 5

The previous chapter does not give a lot of insight into the filter bank structure of BCH codes. Therefore, we have been searching for a more general framework that explains why the techniques as presented before hold. This framework is found in the context of 2-D cyclic codes. It turns out that the roots of the RS/BCH code are spread over the subband codes, which -as we show in Section 5.3- can be seen as BCH codes themselves. The insight gained leads to a number of practical results in Section 5.4. More specifically, recursive filter banks (Section 5.4.1) or unequal error protecting codes (Section 5.4.3) can be constructed starting from the previous theory. However, the insight gained in this chapter is not necessary for further understanding of the text and this chapter can be skipped on a first reading.

Chapter 6

In the second part of this thesis, we look into solutions of the aforementioned problems. Chapter 6 deals with a first application: the filter bank decomposition is used to build a Soft-In Soft-Out (SISO) RS decoder. Following an introduction on optimal decoding criteria (Section 6.2), the MAP criterion is selected and a suboptimal decoding technique called Gallager's algorithm is presented in Section 6.3. In Section 6.4, it is seen how Gallager's algorithm is deployed to make a soft RS decoder using the filter bank decomposition.

The development of an efficient SISO RS decoder is a fundamental open prob-

lem in coding theory, that attracted a lot of attention recently. The filter bank representation developed in Part I is the key concept in finding a suitable decomposition for RS codes. Essentially, the filter bank representations allow us to break one large RS code into a number of subcodes with a sparse interconnect. The subband variables play a central role in this decomposition in order to break the SISO decoder into smaller SISO decoders of the subcodes that exchange information, as dictated by Gallager's algorithm. From a practical perspective, the development of a SISO RS decoder is the main contribution of this thesis. Performance gains up to 1.5 dB have been shown by simulations (in the case of BPSK modulation).

Chapter 7

In chapter 7, a second application is presented. Following a brief introduction on OFDM (Section 7.2), the starting point is an OFDM scheme in which the circulant channel matrix is decomposed using DFT matrices in a Galois field instead of the complex field, called GF-OFDM (Section 7.3). In order to be compatible with the finite field operations, the channel is assumed to be quantized and a modulo operation must be added at the receiver. Using the filter bank representation of an RS code, Section 7.4 explains how this GF-OFDM scheme can be seamlessly merged with an RS code, resulting in a new scheme called RS-OFDM. From a practical point of view, RS-OFDM is a second important contribution. The jointly designed RS-OFDM scheme results in a significant performance gain, in some scenarios up to 10 dB, shown by simulations. In addition to the performance gain, the newly developed RS-OFDM scheme has no Peak-To-Average Power Ratio (PAPR) problem due to the modulo operation. Also the complexity can be lowered since equalization and decoding can be performed by one large RS decoder.

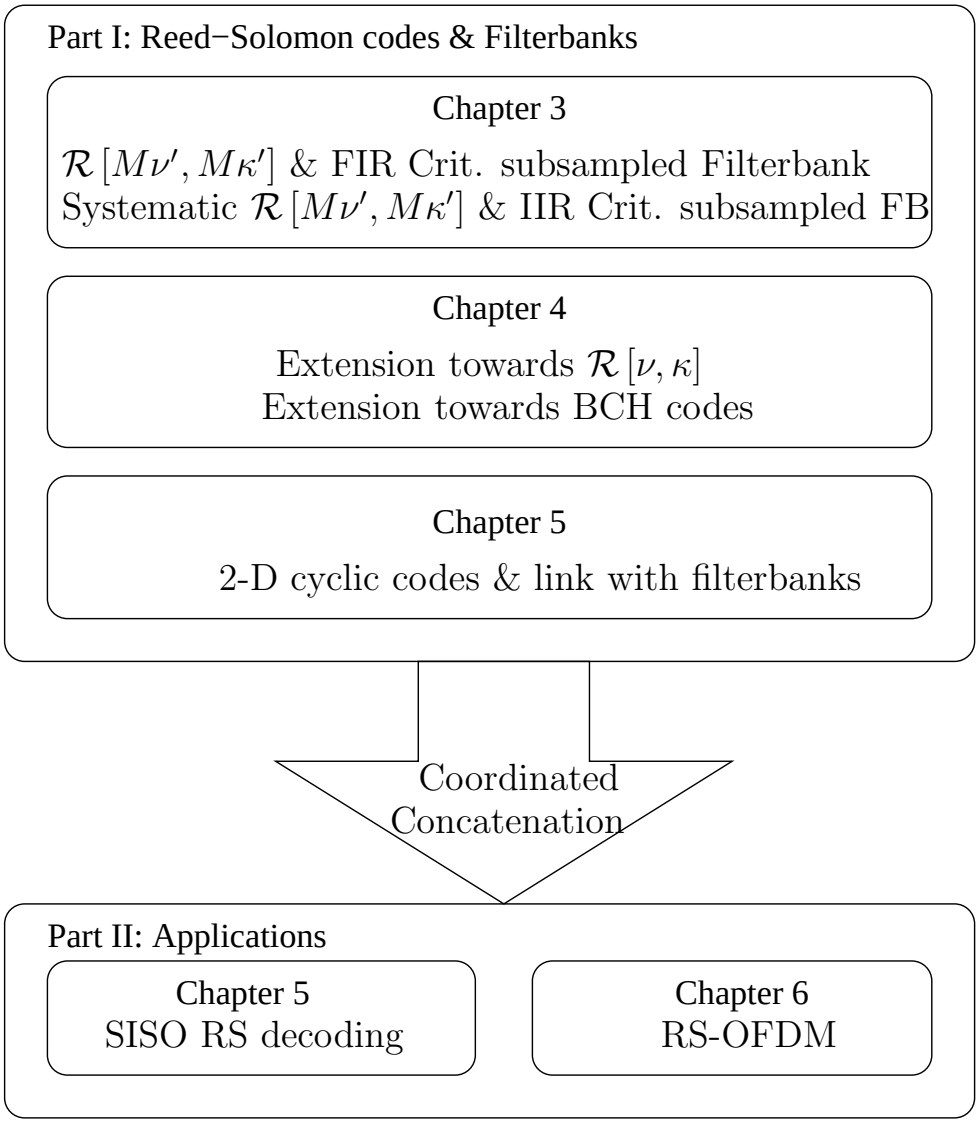


Figure 1.3: Concatenation as a central key message

Chapter 2

Finite Field and Coding Theory

The area of coding theory is rather large, and makes use of many concepts drawn from finite field algebra, which is by itself a vast domain. Instead of giving a formal mathematical description of finite field and coding theory, an intuitive explanation is given. For a more detailed survey, we refer the interested reader to [67, Chapter 3,4,7,8,9,10]. The material is mostly explained from a signal processing point of view. In this respect, the finite field Fourier Transform takes an important role. This transform, which is well known to a signal processing audience, allows us to explain cyclic, BCH and RS codes from a different viewpoint. This approach is closely related to Blahut's paper [8]. Although the finite field Fourier Transform is well known to coding people under another name (Mattson-Solomon polynomial), Blahut realized that it can take a more central role in explaining the topic. Apart from pedagogical reasons, a signal processing point of view at coding theory brings typically new insights, and new relationships can be established between these two domains. The material presented in this thesis is believed to be a good illustration of this.

This chapter consists of two sections. In the first section, finite field theory is briefly introduced. More attention is paid to the finite field counterpart of the well known Discrete Fourier Transform (DFT). This transform plays an important role in the second section which covers basic coding theory. Cyclic codes, a subclass of linear codes, are the main topic of this section. BCH codes are an important subclass of cyclic codes, since the BCH code construction guarantees a certain minimum distance. Finally, RS codes are explained as a special case of BCH codes.

2.1 Finite Field Theory

A field $\mathscr{F}$ is a set of elements in which addition, subtraction, multiplication and division (except by 0) are defined. Addition and multiplication satisfy commutativity, associativity and distributivity. A finite field or Galois field is a field with a finite number of elements. The number of elements, also called the order of the field, is always a prime p or a power of a prime $q = p^m$, $m \in \mathbb{N}_0$. The notation $\mathbb{F}_q$ is used to denote a field with q elements. p is called the characteristic of the field $\mathbb{F}_q$. Up to an isomorphism, each finite field is unique, i.e. there exists no other finite field of the same order. The simplest field is a prime field $\mathbb{F}_p$, consisting of the integers modulo p.

Example 2.1 For example, the finite field $\mathbb{F}_3$ consists of the elements 0, 1 and 2 which satisfy the addition and multiplication tables (Table 2.1). △

Table 2.1: Addition and multiplication tables defining $\mathbb{F}_3$.

+	0	1	2
0	0	1	2
1	1	2	0
2	2	0	1

·	0	1	2
0	0	0	0
1	0	1	2
2	0	2	1

Galois fields of order $q = p^m$ are similarly defined using polynomials: $\mathbb{F}_{p^m}$ consists of all polynomials of degree $m-1$ with coefficients belonging to $\mathbb{F}_p$ modulo an irreducible polynomial of degree m. Often, a primitive polynomial $\mathscr{P}(x)$ is chosen, because its roots generate the field $\mathbb{F}_{p^m}$ (See the example). Recall that *any* irreducible polynomial of degree m yields the same field up to an isomorphism. $\mathbb{F}_{p^m}$ is called an extension field of the (base) field $\mathbb{F}_p$.

Example 2.2 For example, for $\mathbb{F}_{3^2}$, the modulus can be taken as $\mathscr{P}(x) = x^2 + 2x + 2$ or $\mathscr{P}'(x) = x^2 + x + 2$; both are primitive polynomials. Using the modulus $\mathscr{P}(x)$, the elements of $\mathbb{F}_{3^2}$-written 0, 1, x, x^2, ... can be represented as polynomials with degree less than $m = 2$. Now consider Table 2.2 which contains several representations of the elements of a finite field. The columns are the exponential representation, polynomial representation and 2-tuples of polynomial representation coefficients (the vector representation). The vector representation gives rise to another interpretation of the extension field $\mathbb{F}_{p^m}$: $\mathbb{F}_{p^m}$ can be seen as a vector space over $\mathbb{F}_p$ with a basis $\boldsymbol{\lambda}$.

Instead of working with polynomials in x, a root of the primitive polynomial $\mathscr{P}(x)$, called β, is commonly used to denote the Galois field elements. Hence, x can simply be replaced by β in the previous table. Note that β is the 4-th root of 2 mod 3 (row 6 in Table 2.2). So by adjoining the element $\beta \in \mathbb{F}_{3^2}$,

Table 2.2: Three representations of elements in $\mathbb{F}_{3^2}$ defined by $\mathscr{P}(x) = x^2 + 2x + 2$.

vector	polynomial	exponential
$\begin{bmatrix}0 & 0\end{bmatrix}$	0	0
$\begin{bmatrix}1 & 0\end{bmatrix}$	1	1
$\begin{bmatrix}0 & 1\end{bmatrix}$	x	x
$\begin{bmatrix}1 & 1\end{bmatrix}$	$1+1x$	x^2
$\begin{bmatrix}1 & 2\end{bmatrix}$	$1+2x$	x^3
$\begin{bmatrix}2 & 0\end{bmatrix}$	2	x^4
$\begin{bmatrix}0 & 2\end{bmatrix}$	$2x$	x^5
$\begin{bmatrix}2 & 2\end{bmatrix}$	$2+2x$	x^6
$\begin{bmatrix}2 & 1\end{bmatrix}$	$2+x$	x^7

the square root of 2 mod 3 (equal to β^2) and 4-th root of 2 mod 3 (equal to β) exist. Compare this with the complex field $\mathbb{C}$ as an extension field of the real field $\mathbb{R}$ by adjoining the imaginary element j, solution of $\mathscr{P}(x) = x^2 + 1$. $\mathbb{F}_{3^2}$ can also be seen as a vector space over $\mathbb{F}_3$ with a basis $\boldsymbol{\lambda} = \begin{bmatrix}1 & \beta\end{bmatrix} \in \mathbb{F}_{3^2}$. To satisfy linear independence, $\beta \notin \mathbb{F}_3$. △

It can be proven [67, Chapter 4, 2] that the nonzero elements of $\mathbb{F}_q$ form a cyclic multiplicative group of order $q - 1 = p^m - 1$. I.e. $\mathbb{F}_q$ is a multiplicative group consisting of the elements $1, \beta, \beta^2, \ldots, \beta^{q-2}, \beta^{q-1} = 1$. This results leads to Fermat's little theorem:

Theorem 2.1 *(Fermat's little theorem) Every element $a \in \mathbb{F}_q$ satisfies*

$$a^q = a\,.$$

Fermat's little theorem implies that *every* element $a \in \mathbb{F}_q$, $q = p^m$ is a solution of

$$x^q - x = \prod_{a \in \mathbb{F}_q} (x - a) = 0\,. \tag{2.1}$$

As previously mentioned, the procedure of obtaining the field $\mathbb{F}_{p^m}$ from $\mathbb{F}_p$ by adjoining a root of a primitive polynomial can now be iterated. Indeed, defining an irreducible polynomial over $\mathbb{F}_{p^m}$ $\Xi(x)$ of degree n, a root of $\Xi(x)$ can be adjoined to $\mathbb{F}_{p^m}$, resulting in a field $\mathbb{F}_{p^{m\cdot n}} = \mathbb{F}_{q^n}$. This field is also unique. The field $\mathbb{F}_p$ is a subfield of $\mathbb{F}_{p^m}$, denoted as $\mathbb{F}_p \subset \mathbb{F}_{p^m}$. Analogously, $\mathbb{F}_p \subset \mathbb{F}_{p^{m\cdot n}}$ and $\mathbb{F}_{p^m} \subset \mathbb{F}_{p^{m\cdot n}}$. This construction allows us to derive a useful property about subfields.

Theorem 2.2 *$\mathbb{F}_q$ is a subfield of $\mathbb{F}_{q^n}$, $\forall n \in \mathbb{N}_0$.*

2.1.1 Finite Field Transforms

The Fourier transform plays a central role in many signal processing applications. In this work, we explain that its finite field counterpart can be useful to explain error correcting codes from a signal processing point of view. Recall the definition of the Discrete Fourier Transform (DFT) (in the complex field) of a vector $\mathbf{a}$ (also called the spectrum of $\mathbf{a}$) denoted with $\overline{\mathbf{a}}$:

$$\overline{\mathbf{a}}\,[j] = \sum_{i=0}^{\nu-1} W_{\nu}^{ij} a\,[i]\,.$$

with $W_\nu = e^{-\frac{j2\Pi}{\nu}}$, called a ν-th root of unity. Now, let α represent a primitive ν-th root of unity in the Galois field. This α can typically not be found in $\mathbb{F}_q$, but exists in an extension field $\mathbb{F}_{q^n}$[1]. Powers of α form a multiplicative subgroup of $\mathbb{F}_{q^n}$. Equivalently, the finite field DFT is defined as

$$\overline{a}\,[j] = \sum_{i=0}^{\nu-1} \alpha^{ij} a\,[i]\,.$$

The time domain vector $\mathbf{a}$ can be recovered from its spectrum with the inverse operation:

$$\overline{a}\,[j] = \frac{1}{\nu \bmod p} \sum_{i=0}^{\nu-1} \alpha^{-ij} a\,[i]\,.$$

The DFT in the finite field inherits many properties from its non-finite field counterpart, like the convolution property and Parseval's formula. If two time domain signals $\mathbf{a}$ and $\mathbf{b}$ are multiplied (component-wise) resulting in $\mathbf{c}$, the spectrum of $\mathbf{c}$ is obtained through convolution:

$$\overline{c}\,[j] = \frac{1}{\nu} \sum_{i=0}^{\nu-1} \overline{a}\,[j-i]\,\overline{b}\,[i]\,.$$

If polynomials $a(x)$ and $\overline{a}(x)$ are associated with the vectors $\mathbf{a}$ and $\overline{\mathbf{a}}$, an interesting property is easily derived:

$$a(\alpha^j) = \sum_{i=0}^{\nu-1} a\,[i]\,\alpha^{ij} = \overline{a}\,[j]\,.$$

This means that α^j is a root of the polynomial $a(x)$ iff. $\overline{a}\,[j] = 0$. $\overline{a}(x)$ is commonly referred to as the Mattson-Solomon polynomial in coding theory. Hence, in finite fields, speaking of a root of a polynomial or of a spectral component being zero is exactly the same. Only the terminology is different, and the two notions appeal to different audiences.

[1] n can be found as the multiplicative order of q modulo ν.

2.2 Coding Theory

2.2.1 (Non-)Linear Block Codes

Since mainly block codes are considered in this thesis,[2] let us start with a formal definition of a (possibly nonlinear) block code $\mathcal{C}(\nu, M, \delta)$ [67].

Definition 2.3 (Block code) A code $\mathcal{C}(\nu, M, \delta)$ is a set of M vectors (codewords $\mathbf{y}$) of length ν with components from some field $\mathbb{F}_q$ such that any two vectors differ in at least δ places, and δ is the largest number with this property. The rate of the code $\rho = \frac{log_q(M)}{\nu}$. □

Example 2.3 In this example, a code $\mathcal{C}(4, 9, 3)$ is considered. Further on in this thesis, $\mathcal{C}(4, 9, 3)$ is seen to be a unique linear type III[3] self-dual code in $\mathbb{F}_3$ with block length $\nu = 4$ and highest possible Hamming distance $\delta = 3$. The codewords are listed in Table 2.3.

Table 2.3: Codewords of $\mathcal{C}(4, 9, 3)$ in example 2.3

0	0	0	0
0	1	-1	1
0	-1	1	-1
1	1	1	0
1	-1	0	1
1	0	-1	-1
-1	-1	-1	0
-1	0	1	1
-1	1	0	-1

These codewords can be visualized by the tetrahedron (hypercube) shown in Figure 2.1, which can be used to verify the minimum Hamming distance 3.

△

Linear codes are an important subclass since their codewords can be generated relatively simply. From a mathematical point of view, decoding a linear code is significantly simpler than a nonlinear one. For these reasons, nonlinear codes are not dealt with in this dissertation.

[2] The convolutional codes in this thesis have always a predefined length, e.g. by termination, and can therefore be seen as block codes.

[3] The type III refers to the distance which is a multiple of 3 and $\nu|4$ [15].

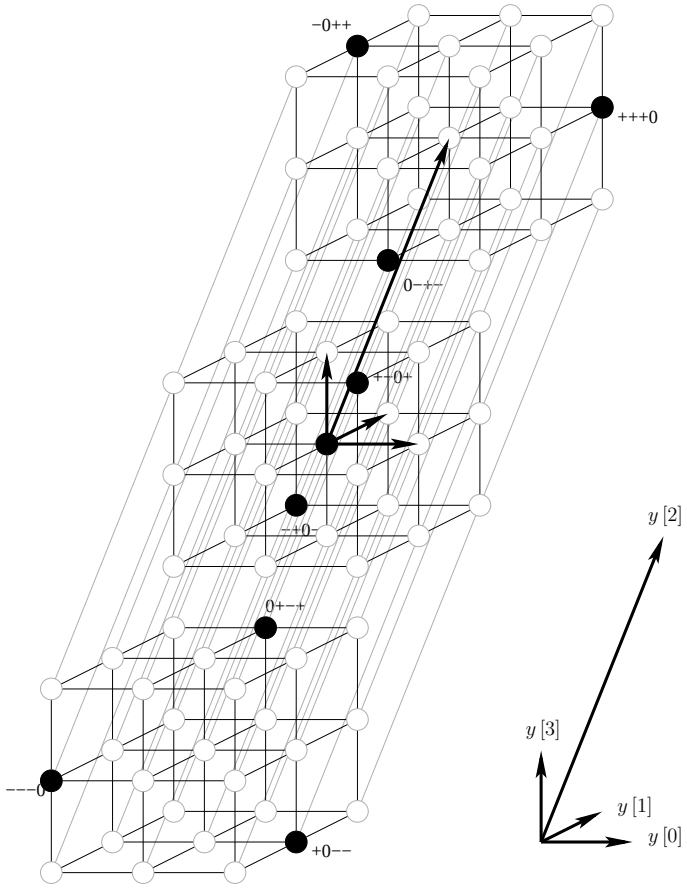


Figure 2.1: Hypercube representing the codewords of the self-dual code $\mathcal{C}(4, 9, 3)$.

Definition 2.4 (Linear code) A linear code of dimension κ and length ν over a field $\mathscr{F}_q$ is a κ-dimensional subspace of an ν-dimensional vector space over $\mathscr{F}_q$. A linear code is denoted with $\mathcal{L}[\nu, \kappa]$. If the minimum distance δ of the code is known, it can be included in the notation as follows: $\mathcal{L}[\nu, \kappa, \delta]$. □

Hence, the code is generated by the columns of an $\nu \times \kappa$ generator matrix $\mathbf{G}_\mathcal{L}$ in $\mathbb{F}_q$. This generator matrix is called *systematic* if $\mathbf{G}_\mathcal{L}[0 : \kappa - 1, 0 : \kappa - 1] = \mathbf{I}_\kappa$. The $\nu - \kappa \times \nu$ matrix

$$\mathbf{H}_\mathcal{L}^T = \text{Null}\,(\mathbf{G}_\mathcal{L}^T)$$

is called the *parity check matrix*. If $\mathbf{y} \in \mathcal{L}$, $\mathbf{H}_\mathcal{L}\mathbf{y} = \mathbf{0}$. Note that the same code $\mathcal{L}$ is generated by $\mathbf{G}_\mathcal{L}^V$, obtained by performing elementary column operations on $\mathbf{G}_\mathcal{L}$ (summarized in the basis transformation matrix $\mathbf{V}$ such that $\mathbf{G}_\mathcal{L} = \mathbf{G}_\mathcal{L}^V\mathbf{V}$). Hence, the codebook of the codes generated by $\mathbf{G}_\mathcal{L}^V$ and $\mathbf{G}_\mathcal{L}$ is the same. This leads to the important notion of equivalent codes: if the codebook of two codes differs only in the order of the symbols, the codes are called *equivalent*.

An important property of a linear code is that its (Hamming) distance corresponds to the minimum weight of a non-zero codeword. So the distances between codewords should never be checked. A linear code $\mathcal{L}[\nu, \kappa]$ has distance δ if every $\delta - 1$ columns of $\mathbf{H}_\mathcal{L}$ are linearly independent while at least one set of δ columns can be found which is linearly dependent. Since the maximum number of linearly independent columns of $\mathbf{H}_\mathcal{L}$ is $\nu - \kappa$, a simple upper bound on the maximum Hamming distance can be obtained:

Theorem 2.5 *(Singleton bound) The maximum Hamming distance of a linear code $\mathcal{L}[\nu, \kappa, \delta]$ is $\nu - \kappa + 1$.*

Codes for which $\delta = \nu - \kappa + 1$ are called *Maximum Distance Separable* (MDS) codes.

Example 2.4 The code $\mathcal{C}(4, 9, 3)$ of Example 2.3 can be verified to be a linear code in $\mathbb{F}_3$. Hence, it will be denoted as $\mathcal{L}[4, 2, 3]$. With $\kappa = 2$ and $\nu = 4$, the generator and parity check matrices are

$$\begin{aligned} \mathbf{G}_\mathcal{L} &= \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 1 & -1 \\ 0 & 1 \end{bmatrix} \\ \mathbf{H}_\mathcal{L} &= \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & -1 & 1 \end{bmatrix} \end{aligned}$$

△

2.2.2 Cyclic Codes

Cyclic codes (or Ideals) belong to the class of linear codes.

Definition 2.6 (Cyclic code or Ideal) A linear code is called cyclic if each cyclic shift of a codeword results in an other codeword:

$$\mathbf{y} = \begin{bmatrix} y(0) & y(1) & \dots & y(\nu-1) \end{bmatrix} \in \mathcal{I}[\nu,\kappa] \Rightarrow \\ \begin{bmatrix} y(\nu-1) & y(0) & y(1) & \dots & y(\nu-2) \end{bmatrix} \in \mathcal{I}[\nu,\kappa]$$

□

Cyclic codes are conventionally described in terms of polynomials. There is a unique codeword $g_{\mathcal{I}}(x)$ (up to a scalar) that generates the whole code $\mathcal{I}[\nu,\kappa]$, and is therefore called the generator polynomial. A few other properties concerning $g_{\mathcal{I}}(x)$ are that $g_{\mathcal{I}}(x)|x^{\nu}-1$ and that each codeword $y(x)$ can be written uniquely as

$$y(x) = g_{\mathcal{I}}(x)u(x)\,. \tag{2.2}$$

With the dataword length equal to κ, the degree of $g_{\mathcal{I}}(x)$ equals $\nu-\kappa$. Hence, the code is generated by the columns of a $\nu \times \kappa$ generator matrix $\mathbf{G}_{\mathcal{I}}$. This matrix has a Toeplitz structure:

$$\mathbf{G}_{\mathcal{I}} = \begin{bmatrix} g_{\mathcal{I}}[0] & & \\ g_{\mathcal{I}}[1] & \ddots & \\ \vdots & & g_{\mathcal{I}}[0] \\ g_{\mathcal{I}}[\nu-\kappa] & & g_{\mathcal{I}}[1] \\ & \ddots & \vdots \\ & & g_{\mathcal{I}}[\nu-\kappa] \end{bmatrix}.$$

Since $g_{\mathcal{I}}(x)|x^{\nu}-1$, the polynomial $h(x) = \frac{(x^{\nu}-1)}{g_{\mathcal{I}}(x)}$ exists, referred to as the check polynomial. The name refers to the fact that a multiplication $y(x)h(x) = 0$ iff. $y(x) \in \mathcal{I}[\nu,\kappa]$. The parity check matrix is the nullspace of the generator matrix $\mathbf{G}_{\mathcal{I}}$ and has also a Toeplitz structure:

$$\mathbf{H}_{\mathcal{I}} = \begin{bmatrix} h_{\mathcal{I}}[\kappa] & h_{\mathcal{I}}[\kappa-1] & \dots & h_{\mathcal{I}}[0] & & \\ & \ddots & & & \ddots & \\ & & h_{\mathcal{I}}[\kappa] & h_{\mathcal{I}}[\kappa-1] & \dots & h_{\mathcal{I}}[0] \end{bmatrix}.$$

Example 2.5 In this example, a cyclic code of length $\nu = 10$ in $\mathbb{F}_{3^2}$ is constructed. The primitive polynomial $\mathscr{P}(x) = 2 + 2x + x^2$ (See Table 2.2). We start by factoring

$$x^{10}-1=\underbrace{(x-1)(x+1)(1+\beta x+x^2)(1+\beta^3 x+x^2)}_{h_{\mathcal{I}}(x)}$$
$$\underbrace{(1+\beta^5 x+x^2)(1+\beta^7 x+x^2)}_{g_{\mathcal{I}}(x)}.$$

The check polynomial consists of the first 4 factors, while the generator polynomial is composed of the last 2 factors. The generator matrix $\mathbf{G}_{\mathcal{I}}$ is given by

$$\mathbf{G}_{\mathcal{I}}=\begin{bmatrix}
1 & & & & & \\
\beta^4 & 1 & & & & \\
1 & \beta^4 & 1 & & & \\
\beta^4 & 1 & \beta^4 & 1 & & \\
1 & \beta^4 & 1 & \beta^4 & 1 & \\
 & 1 & \beta^4 & 1 & \beta^4 & 1 \\
 & & 1 & \beta^4 & 1 & \beta^4 \\
 & & & 1 & \beta^4 & 1 \\
 & & & & 1 & \beta^4 \\
 & & & & & 1
\end{bmatrix}$$

. The minimum distance of this code is only $\delta=2$. △

2.2.3 Cyclic codes in the Frequency Domain

Equation 2.2 is a convolution in the time domain. In the frequency domain, this convolution translates into a (component-wise) product

$$\overline{y}[j]=\overline{g}[j]\,\overline{u}[j],\forall j=0:\nu-1.$$

Note that time-domain vectors have elements in $\mathbb{F}_q$ while frequency-domain vectors have elements in an extension field $\mathbb{F}_{q^n}$. Because the signal spectrum $\overline{\mathbf{u}}$ is arbitrary, the only significant role of $\overline{\mathbf{g}}$ is to put a number of zeros at certain frequencies in the codeword spectrum. These so called parity check frequencies correspond with the zeros of the generator polynomial $g_{\mathcal{I}}(x)$. Hence, a cyclic code can be seen as a set of ν-point inverse DFT transforms of all spectral vectors that are zero at certain prescribed frequencies. However, it is not possible to choose just any spectrum that is zero in the parity check frequencies. Some of these might have inverse DFT transforms of which the elements do not belong to $\mathbb{F}_q$. So, the non-parity check frequencies must be chosen to represent the information only in those q^κ possible ways that correspond to $\mathbb{F}_q$-valued inverse transforms. Hence, well defined constraints are aimed for, that allow us to choose the remaining positions in such a way that the codeword belongs to $\mathbb{F}_q$. These constraints are called conjugacy constraints:

Theorem 2.7 *(Conjugacy constraints) Let* $\overline{\mathbf{y}}$ *be a spectrum of a* $\nu\times 1$ *vector* $\mathbf{y}$. $\mathbf{y}\in\mathbb{F}_q$ *iff.*

$$\overline{y}[j]^q=\overline{y}[qj \bmod \nu]\ .$$

This theorem allows us to divide the integers modulo ν into sets, called cyclotomic cosets:

Definition 2.8 (Cyclotomic coset) The cyclotomic coset $\mathscr{C}_{q,\nu}^{(s)}$ over $\mathbb{F}_q$ modulo ν containing an integer s is the set

$$\mathscr{C}_{q,\nu}^{(s)} = \{q^k s \bmod \nu\}_{k=0..m_s-1}$$

with m_s the smallest positive integer such that $q^{m_s} = s \bmod \nu$. s is the smallest number in the coset referred to as the *coset representative*. □

So if the spectral component $\overline{\mathbf{y}}[j]$ is specified, all other spectral components with indices in the cyclotomic coset must be a power of $\overline{\mathbf{y}}[j]$. Moreover, since $\overline{\mathbf{y}}[j]^{q^{m_s}} = \overline{\mathbf{y}}[j]$, we are not free to choose any element of $\mathbb{F}_{q^n}$ for $\overline{\mathbf{y}}[j]$, but only those of order $q^{m_s} - 1$.

Example 2.6 The 10-th roots of unity (solutions of $x^{10} - 1$) exist in $\mathbb{F}_{3^4}$ since the multiplicative order of 9 mod 10 ($q \bmod \nu$) equals $n = 2$. The primitive polynomial defining $\mathbb{F}_{3^4}$ is $\mathscr{P}(x) = 2 + 2x^3 + x^4$, with root γ. $\mathbb{F}_{3^2}$ is a subfield of $\mathbb{F}_{3^4}$, and the primitive element $\beta \in \mathbb{F}_{3^2}$ equals γ^{10}. The 10-th roots of unity are $\alpha^0 = \gamma^0, \alpha^1 = \gamma^8, \ldots, \alpha^9 = \gamma^{72}$. The cyclotomic cosets over $\mathbb{F}_{3^2}$ modulo 10 can be found in Table 2.4. The cyclic code $\mathcal{I}[10, 6, 2]$ of example 2.5 is obtained by $g_{\mathcal{I}}(x) = \prod_k (x - \alpha^k), \forall k \in \begin{bmatrix} 1 & 3 & 6 & 7 \end{bmatrix}$. Figure 2.2 illustrates

Table 2.4: Cyclotomic cosets over $\mathbb{F}_{3^2}$ modulo 10

Cyclotomic coset	Roots
$\mathscr{C}_{9,10}^{(0)} = \{\ 0\ \}$	α^0
$\mathscr{C}_{9,10}^{(1)} = \{\ 1 \quad 9\ \}$	α^1, α^9
$\mathscr{C}_{9,10}^{(2)} = \{\ 2 \quad 8\ \}$	α^2, α^8
$\mathscr{C}_{9,10}^{(3)} = \{\ 3 \quad 7\ \}$	α^3, α^7
$\mathscr{C}_{9,10}^{(4)} = \{\ 4 \quad 6\ \}$	α^4, α^6
$\mathscr{C}_{9,10}^{(5)} = \{\ 5\ \}$	α^5

the relations between the various fields. Let us also illustrate frequency domain encoding. The following spectrum obeys the parity check constraints since it is zero at the locations $\begin{bmatrix} 1 & 3 & 6 & 7 \end{bmatrix}$:

$$\overline{\mathbf{y}} = \begin{bmatrix} \gamma^{20} & 0 & \gamma^{74} & 0 & \gamma^{60} & 1 & \gamma^{60} & 0 & \gamma^{26} & 0 \end{bmatrix}$$

Also the conjugacy constraints are obeyed:

$$y[8] \quad = \quad y[9 \cdot 2 \bmod 10] = y[2]^9 = \gamma^{26}$$

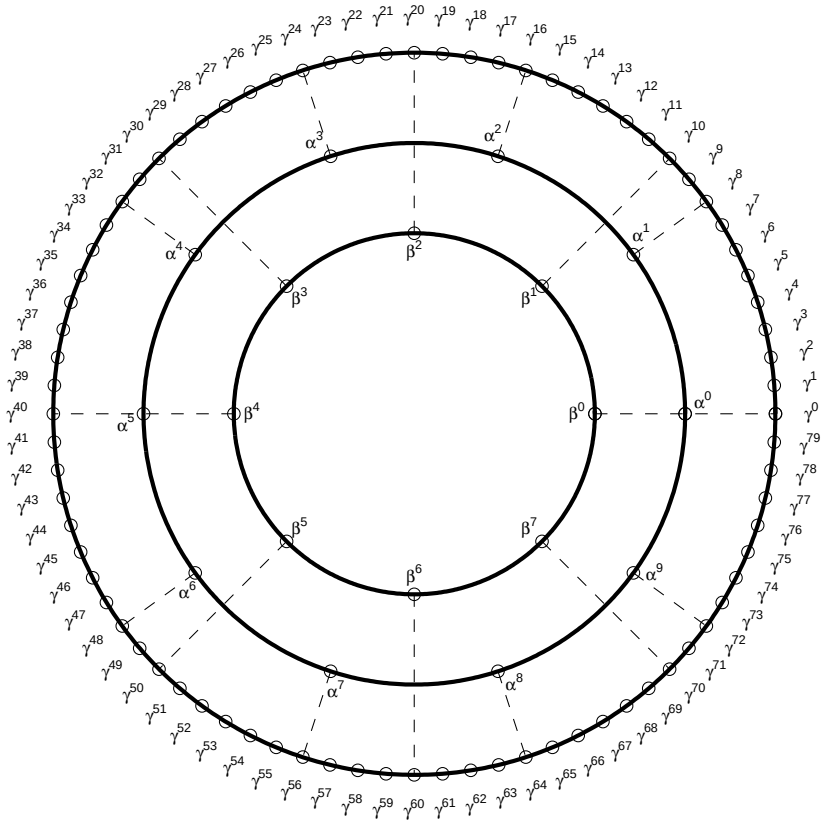


Figure 2.2: Relations between primitive elements $\beta \in \mathbb{F}_{3^2}$, $\gamma \in \mathbb{F}_{3^4}$ and the 10-th roots of unity α.

$$y[6] \quad = \quad y[9 \cdot 4 \bmod 10] = y[4]^9 = \gamma^{60}$$

The elements at these positions should be of order 80, and can therefore freely be chosen from $\mathbb{F}_{3^4}$. However, $y[0]$ and $y[5]$ should have order 8 (belong to $\mathbb{F}_{3^2}$) to ensure that the codeword (in the time domain) belongs to $\mathbb{F}_{3^2}$. An inverse DFT operation results in the codeword

$$\mathbf{y} = \left[\begin{array}{cccccccccc} \beta^7 & 1 & 1 & \beta^5 & \beta^6 & \beta^1 & 0 & \beta^4 & \beta^6 & \beta^5 \end{array} \right].$$

△

2.2.4 BCH and RS Codes

This section deals with a special type of cyclic codes, named *Bose-Chaudhuri-Hochquenghem* (BCH) code after their inventors.

Definition 2.9 (BCH code) A BCH code $\mathcal{B}[\nu, \kappa]$ over $\mathbb{F}_q$ of length ν and designed distance d is the largest possible cyclic code over $\mathbb{F}_q$ with zeros

$$\alpha^b, \alpha^{b+1}, \ldots, \alpha^{b+d-2}.$$

α is a primitive ν-th root of unity in the extension field $\mathbb{F}_{q^n}$. □

If $\nu = q^m - 1$, $\mathcal{B}[\nu, \kappa]$ is said to be a *primitive BCH code*. If $b = 1$, $\mathcal{B}[\nu, \kappa]$ is called a *narrow sense BCH code*. Due to the specific construction of BCH codes, a statement can be made about the minimum distance of the code:

Theorem 2.10 *(BCH bound) The true minimum distance δ of the BCH code $\mathcal{B}[\nu, \kappa]$ is at least as large as the designed distance d.*

The surprisingly simple proof can be found in [67, Ch. 7, §6].

Example 2.7 Continuing Example 2.6, let us construct a cyclic code with a weight larger than 2, e.g. a narrow sense BCH code with designed distance $d = 3$. Therefore at least 2 consecutive powers of α must be chosen: α, α^2. This results in a generator polynomial

$$g_{\mathcal{B}}(x) = 1 + \beta^6 x + \beta^6 x^3 + x^4. \tag{2.3}$$

The distance of this BCH code can be verified to be $\delta = 4 > d$. △

Is there still room for improvement? The Singleton bound (Theorem 2.5) suggests $\delta = 5$ might exist. To achieve this bound, a set of cyclotomic cosets must be picked which yield consecutive powers of α. In this case, an MDS $\mathcal{B}[\nu, \kappa, \delta = \nu - \kappa + 1]$ is obtained with $\delta - 1$ consecutive roots.

Example 2.8 Assume a set of roots is chosen which is symmetric around 1, e.g. $\alpha^{-2} = \alpha^8, \alpha^{-1} = \alpha^9, \alpha^0, \alpha^1, \alpha^2$. With

$$g_{\mathcal{B}}(x) = \beta^4 + \beta^7 x + \beta^6 x^2 + \beta^2 x^3 + \beta^3 x^4 + x^5$$

an MDS BCH code $\mathcal{B}[10, 5, 6]$ is obtained. Such a BCH code is called an *optimal BCH code*. △

The previous example indicates that optimal BCH codes only exist for certain combinations of ν and κ. However, if the length of the code $\nu = q - 1$, the

extension field equals $\mathbb{F}_q$ ($n = 1$) with $\beta = \alpha$. As a consequence, the cyclotomic cosets are singletons: $C_{q,n}^{(s)} = \{s\}$ and the conjugacy constraints are automatically fulfilled. Then, optimal BCH codes exist for each combination of ν and κ. The codes so obtained are called *Reed-Solomon (RS) codes*.

Definition 2.11 (Reed-Solomon code) A Reed-Solomon (RS) code $\mathcal{R}[\nu, \kappa]$ over $\mathbb{F}_q$ is an optimal BCH code of length $\nu = q - 1$. Its designed distance equals the actual distance $d = \delta = \nu - \kappa + 1$. Consequently, all RS codes are MDS codes. □

Example 2.9 An example is given of an RS code of length $\nu = 8$ in $\mathbb{F}_{3^2}$ with primitive polynomial $\mathscr{P}(x) = 2 + 2x + x^2$, defining the field in Table 2.2. A generator polynomial of $\mathcal{R}[8, 6, 3]$

$$g_{\mathcal{R}}(x) = \prod_{i=b}^{b+\delta-2} (x - \alpha^i) = 1 + \alpha^7 x + \alpha^3 x^2,$$

corresponding to the generator matrix

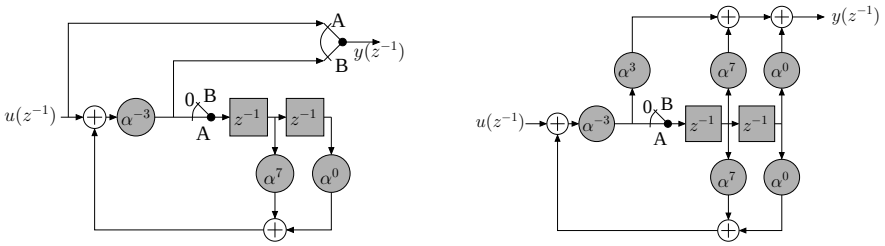
$$\mathbf{G}_{\mathcal{R}} = \begin{bmatrix} \alpha^3 & & & & & & \\ \alpha^7 & \alpha^3 & & & & & \\ 1 & \alpha^7 & \alpha^3 & & & & \\ & 1 & \alpha^7 & \alpha^3 & & & \\ & & 1 & \alpha^7 & \alpha^3 & & \\ & & & 1 & \alpha^7 & \alpha^3 & \\ & & & & 1 & \alpha^7 & \alpha^3 \\ & & & & & 1 & \alpha^7 \\ & & & & & & 1 \end{bmatrix}.$$

△

Encoding of BCH Codes

The encoding process can be easily understood from a signal processing point of view. Due to the Toeplitz character of the BCH code generator matrices, a BCH code $\mathcal{B}[\nu, \kappa]$ (and every cyclic code in general) can be encoded with a terminated *Finite Impulse Response* (FIR) filter $g_{\mathcal{B}}(x)$. In the signal processing community, filters are often denoted in the Z-domain, where x is replaced by z^{-1}. However simple, this encoding can impose some problems. Imagine the decoder obtains a codeword which is correct in all positions but one. Retrieving the dataword by solving the over determined system is impossible. If the dataword is found by keeping κ equations (containing the erroneous one) might result in a dataword that is completely wrong.

This problem can be overcome by so called *systematic* encoding, in which the dataword is embedded in the codeword. In systematic form encoding of BCH



(a) Typical implementation with polynomial divisor (b) Implementation with terminated IIR filter

Figure 2.3: Filter implementations of systematic encoder of $\mathcal{R}[8, 6, 3]$.

codes, each codeword can be written as

$$y(z^{-1}) = u(z^{-1}) + z^{-\kappa}p(z^{-1}). \tag{2.4}$$

The problem is to calculate the parity symbols $p(z^{-1})$ such that $y(z^{-1})$ is a valid codeword. Hence, $y(z^{-1})$ should be a multiple of $g(z^{-1})$ and so $p(z^{-1})$ is found as minus the remainder of $u(z^{-1})z^{\kappa}$ divided by $g(z^{-1})$, i.e.

$$p(z^{-1}) = -u(z^{-1})z^{\kappa} \bmod g(z^{-1}). \tag{2.5}$$

Then $g(z^{-1})q(z^{-1}) = u(z^{-1})z^{\kappa} - p(z^{-1})$ (with $q(z^{-1})$ the quotient) and thus $y(z^{-1}) = g(z^{-1})q(z^{-1})z^{-\kappa}$ is indeed a multiple of $g(z^{-1})$. A signal flow graph implementing a systematic encoder for $\mathcal{R}[8, 6, 3]$ code (example 2.9) can now easily be found (see also [67] and Figure 2.3(a)): in the first stage A (κ clock cycles), the dataword $u(z^{-1})$ is connected with the output and at the same time fed into a polynomial divisor circuit $\frac{1}{g(z^{-1})}$. In the second stage B ($\nu - \kappa$ clock cycles), zeros are fed into the delay line, such that the remainder $p(z^{-1})$ appears at the output. Note that a completely equivalent circuit is shown in Figure 2.3(b). The feed forward filter equals the feedback filter, such that the input is fed through unaltered in stage A yielding the systematic part. In the second stage, the trellis is terminated by feeding zeros in the delay line. Note that the termination of the IIR filter adds the redundancy, which is essentially the same as for the non-systematic encoder (apart from the FIR filter used).

The generator matrix in its systematic form $\mathbf{G}_{\mathcal{B}}^{sys}$ can easily be obtained from

$$\mathbf{G}_{\mathcal{B}}^{sys} = \mathbf{G}_{\mathcal{B}}\mathbf{P}^{-1} \text{ with } \mathbf{P} = \mathbf{G}_{\mathcal{B}}[0 : \kappa - 1, 0 : \kappa - 1]. \tag{2.6}$$

Note that the systematic code generated by $\mathbf{G}_{\mathcal{B}}^{sys}$ is equivalent to its non-systematic counterpart generated by the Toeplitz matrix $\mathbf{G}_{\mathcal{B}}$: both share the same codebook.

Part I

Reed-Solomon Codes and Filter Banks

Chapter 3

Reed-Solomon Codes and Filter Banks

3.1 Introduction

Filter banks have long been known to be a powerful tool for image and audio processing. They are used in DSP applications (to split a long filter into many smaller filters at a lower rate), in source coding, etc. Recently, filter banks have also found their way into communications. Scaglione *et al.* [62] show that many modulation schemes including OFDM, DMT, TDMA, and CDMA can actually be viewed as filter banks that build input diversity (add redundancy) at the transmitter. This dissertation focuses on filter banks that add redundancy in a similar fashion, but now with the goal of error correction.

Also in the literature, some papers can be found that present a filter bank-like scheme that adds redundancy for the purpose of error correction, see e.g. Fekri *et al.* [26, 25]. Fekri uses filter banks and finite field wavelets to study cyclic codes. Using frequency domain constraints, wavelet codes are obtained. A fundamental difference between Fekri's work and this work is the structure of the filter banks. In [26, 25], a filter bank implementing a rate $\frac{K}{L}$ error correcting code shows an analysis bank with a downsampling factor K, while the upsampling factor equals L in the synthesis bank. In the half-rate wavelet encoder [26], the downsampling factor in the analysis bank is 1. To construct a rate $\frac{\kappa}{\nu}$ MDS code, the BCH criterion is called upon to obtain a synthesis bank which puts $\nu - \kappa$ consecutive zeros in the output. In the literature, also oversampled filter banks are used to add redundancy with the purpose of error-correction, see e.g. Labeau *et al.* [45].

This dissertation deals with Reed-Solomon codes, for which a critically subsampled filter bank representation is developed. These filter banks have a completely different structure compared to those described above. The downsample and upsample factor are *equal*. The redundancy arises *exclusively* from the subband filters. The subband filters act as terminated convolutional codes. To obtain low complexity decoding algorithms, the (constraint) length of these subband filters should be small. In this chapter, we will show that the constraint length of the subband codes corresponds to the constraint length of the RS code divided by the number of bands. This is a very interesting property, which allows us to build Soft-In Soft-Out decoders later on. The main purpose of this chapter is to provide insight into the filter bank structure of RS codes. The methods in this chapter can only be applied to RS codes for which the block length ν and κ are not coprime. This constraint will be relaxed in Chapter 4.

Since RS codes are essentially FIR filters, FIR filtering using filter banks is studied first in this chapter (Section 3.2). The main tool called upon is *Cook-Toom's algorithm* [9], which belongs to the class of fast FIR filtering algorithms. Cook-Toom's algorithm is the foundation of fast FIR filtering using overlap-add/save (OLA/OLS) filter banks (See also Vetterli [78]). Starting from these non-critically subsampled filter banks, this chapter gradually evolves towards a critically subsampled filter bank. In Section 3.3, a non-critically subsampled filter bank is constructed with nonzero order subband filters (subcodes). Given this insight, and using the cyclic properties of an RS code, a critically subsampled bank is obtained in Section 3.4. The latter filter bank representation of an RS code is seen to break the RS code into a number of concatenated codes (in parallel and series). Therefore, we also refer to this as the filter bank *decomposition* of an RS code. Finally, in Section 3.5, it is seen how an IIR version of the filter bank is able to implement a systematic RS code.

3.2 Cook-Toom's Algorithm: the Basis of a Filter Bank Implementation

Straightforwardly computing a product $f(z^{-1})$ of two polynomials $g(z^{-1}) = g[0] + g[1]z^{-1} + \ldots + g[L-1]z^{-L+1}$ and $h(z^{-1}) = h[0] + h[1]z^{-1} + \ldots + h[N-1]z^{-N+1}$ requires NL multiplications. The application of *Cook-Toom's algorithm* is known to reduce the number of multiplications to $M \geq N+L-1$. The procedure is as follows:

- Choose M interpolation points $\boldsymbol{\rho}$ that are the roots of

$$r(z^{-1}) = \prod_{i=0}^{M-1} \left(z^{-1} - \rho[i]\right).$$

- Evaluate $f(\rho\,[i]) = g(\rho\,[i])h(\rho\,[i])$ for all i
- Perform *Lagrange* interpolation

$$f(z^{-1}) = \sum_{i=0}^{M-1} f(\rho\,[i])L_i(z^{-1})$$

with

$$L_i(z^{-1}) = \frac{\prod_{k\neq i}(z^{-1} - \rho\,[k])}{\prod_{k\neq i}(\rho\,[i] - \rho\,[k])}\,.$$

Cook-Toom's algorithm is often applied to reduce the computational cost by carefully selecting the roots $\rho\,[i]$, implying a cheap computation of some of the quantities involved here, so that the main computational cost is indeed in the M multiplications $f(\rho\,[i]) = g(\rho\,[i])h(\rho\,[i])$. Also note that because of the underlying polynomial arithmetic, *Cook-Toom's algorithm* is applicable in any field, e.g. in $\mathbb{C}$ but also in the Galois Field $\mathbb{F}_q$. However, our interest goes to the corresponding matrix factorization.

Example 3.1 Let us illustrate Cook-Toom's algorithm with an example in $\mathbb{R}$. The product of $g(z^{-1}) = g\,[0] + g\,[1]\,z^{-1}$ $(L = 2)$ and $h(z^{-1}) = h\,[0] + h\,[1]\,z^{-1}$ $(N = 2)$ is calculated. With the interpolation points chosen to be $\{0, 1, -1\}$, $f(0) = h\,[0]\,g\,[0]$, $f(1) = (h\,[0] + h\,[1])(g\,[0] + g\,[1])$, $f(-1) = (h\,[0] - h\,[1])(g\,[0] - g\,[1])$. 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 $f(z^{-1}) = f(0)L_0(z^{-1}) + f(1)L_1(z^{-1}) + f(-1)L_{-1}(z^{-1})$ is reconstructed. △

If the polynomial multiplication is written as $\mathbf{f} = \mathbf{Gh}$, then *Cook-Toom's algorithm* can be viewed as a matrix decomposition.

Theorem 3.1 *(Cook-Toom's decomposition) Let* $\mathbf{G}$ *be an* $(N + L - 1) \times N$ *Toeplitz matrix defining the filter* $g(z^{-1})$*:*

$$\mathbf{G} = \begin{bmatrix} g\,[0] & & \\ g\,[1] & \ddots & \\ \vdots & & g\,[0] \\ g\,[L-1] & & g\,[1] \\ & \ddots & \vdots \\ & & g\,[L-1] \end{bmatrix}. \tag{3.1}$$

Using Cook-Toom's algorithm leads to the following decomposition

$$\mathbf{G} = \mathbf{CDA}, \tag{3.2}$$

with $\mathbf{D} = Diag(\mathbf{Bg})$. $\mathbf{A}$ *is an* $M \times N$ *matrix with* $A[m,n] = \rho[m]^n$ $(n = 0 : N-1)$, *so that* $\mathbf{Ah}$ *is representative of computing* $h(\rho[m])$, $m = 0 : M-1$. $\mathbf{B}$ *is an* $M \times L$ *matrix with* $\mathbf{B}_{m,l} = \rho[m]^l$ $(l = 0 : L-1)$ *and* $\mathbf{C}$ *is an* $(N+L-1) \times M$ *with the* i*-th column containing the first* $N + L - 1$ *coefficients of* $L_i(z^{-1})$.

Example 3.2 Continuing Example 3.1, we obtain:

$$\mathbf{f} = \underbrace{\begin{bmatrix} g[0] & \\ g[1] & g[0] \\ & g[1] \end{bmatrix}}_{\mathbf{G}} \mathbf{h} = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{2} & -\frac{1}{2} \\ -1 & \frac{1}{2} & \frac{1}{2} \end{bmatrix}}_{\mathbf{C}} \mathbf{D} \underbrace{\begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 1 & -1 \end{bmatrix}}_{\mathbf{A}} \mathbf{h} \tag{3.3}$$

with

$$\mathbf{D} = \mathrm{Diag}\left(\underbrace{\begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 1 & -1 \end{bmatrix}}_{\mathbf{B}} \begin{bmatrix} g[0] \\ g[1] \end{bmatrix} \right) \tag{3.4}$$

. △

At this point, a number of remarks are appropriate.

- First, if Cook-Toom's decomposition of $g(z^{-1}) = 1$ ($L = 1$, $M = N$) is considered, it is easily seen that $\tilde{\mathbf{A}} = \mathbf{C}^{-1}$, with $\tilde{\mathbf{A}}$ an extended version of $\mathbf{A}$ and hence also a Vandermonde matrix:

  $$\mathbf{C}^{-1}[m,n] = \tilde{A}[m,n] = \rho[m]^n, (n = 0 : M-1) \tag{3.5}$$

  Therefore, Cook-Toom's decomposition can be formulated as follows:

  $$\mathbf{G} = \tilde{\mathbf{A}}^{-1}\mathbf{D}\tilde{\mathbf{A}}[0 : M-1, 0 : N-1]. \tag{3.6}$$

- Secondly, note that the same $\mathbf{f}$ can be obtained by multiplying $g_\delta(z^{-1}) = z^{-\delta}g(z^{-1})$ and $h_\delta(z^{-1}) = z^{\delta}h(z^{-1})$. This results in a somewhat different matrix decomposition where $A[m,n] = \rho[m]^{n+\delta}$ and $B[m,l] = \rho[m]^{l-\delta}$. This will be used at the end of the chapter. Unless mentioned otherwise, $\delta = 0$.

- Finally note that the *Cook-Toom's algorithm* is a special case of the *Chinese Remainder Theorem* [9], and that if the degree of $r(z^{-1})$ is too low ($M < L + N - 1$), the convolution modulo $r(z^{-1})$ is calculated:

  $$f(z^{-1}) = g(z^{-1})h(z^{-1}) \bmod r(z^{-1}) \tag{3.7}$$

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

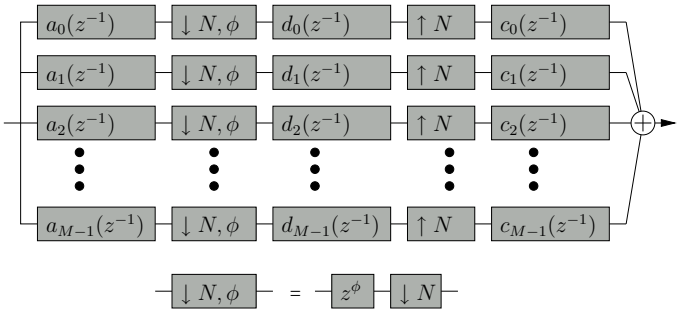


Figure 3.1: An *overlap-add* filter bank with M bands and N-fold subsampling. The filter weights are the diagonal elements of $\mathbf{D}$ in Theorem 3.2.

Theorem 3.2 *Given a decomposition* **CDA** *based on Cook-Toom's algorithm implementing a multiplication with an* $N + L - 1 \times N$ *Toeplitz matrix* $\mathbf{G}$ *(convolution with a filter* $g(z^{-1})$ *of length* L*) using* $M \geq L + N - 1$ *roots. Now, construct a multirate filter bank with* M *bands, indexed by* m*, and subsampled by* N*, as shown in Figure 3.1. The analysis and the synthesis filters of band* $m = 0 : M - 1$ *are defined as follows:*

$$a_m(z^{-1}) = \sum_{k=0}^{N-1} A\left[m, N-1-k\right] z^{-k} \tag{3.8}$$

$$c_m(z^{-1}) = \sum_{k=0}^{L+N-2} C\left[k, m\right] z^{-k} \tag{3.9}$$

With the diagonal elements of $\mathbf{D}$ *residing in the subband filters, this filter bank will implement a so-called overlap-add convolution with* $z^{-N+1+\phi} g(z^{-1})$*, with* ϕ *the phase of the downsamplers (*$0 : N - 1$*).*

The *overlap-add* scheme is widely described in literature [56, 79]. The filter bank splits a large data-block into small blocks of length N, with the first block containing $\phi+1$ samples. This ϕ represents the phase of the downsamplers and is indicated as the second number that can be found in a downsample block of a figure. Note that if $\phi > 0$, the downsamplers implement a non-causal operation, which is typically not a problem for the receiver since it is block oriented anyway.

For each small block of N samples, the convolution with $g(z^{-1})$ is calculated, based on the decomposition **CDA**. This is illustrated in the upper part of Figure 3.2. 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. Finally, in the synthesis bank represented by a multiplication with $\mathbf{C}$, the overlapping parts of length $L - 1$ are added, which explains the name.

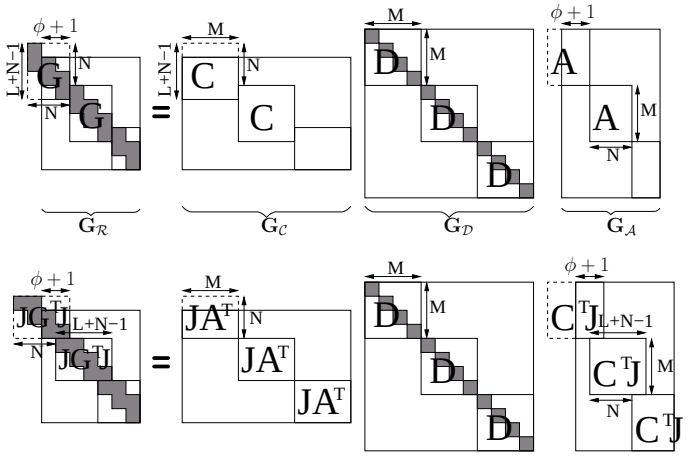


Figure 3.2: Filter bank from a matrix decomposition point of view. The *overlap-add* and the *overlap-save* versions are shown respectively in the upper and the lower parts of the figure. The phase ϕ of the downsamplers is also indicated.

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*.

It is noted that signal flow transposition can be applied to obtain the overlap-save variant. The large Toeplitz matrix is then broken into fat Toeplitz matrices that are equal to $\mathbf{JG}^T\mathbf{J}$ *with* $\mathbf{J}$ *the exchange matrix (ones on the anti-diagonal). See also the lower part of Figure 3.2. The* $\mathbf{CDA}$ *decomposition is transposed, and is left and right-multiplied with* $\mathbf{J}$*. The resulting (overlap-save) filter bank is exactly the same as its overlap-add equivalent, but with synthesis and analysis bank swapped.*

Example 3.3 In this example, let $\mathcal{R}[15, 10]$ represent an RS code over $\mathbb{F}_{2^4}$, defined by the primitive polynomial $\mathscr{P}(x) = x^4 + x + 1$; α a primitive 15-th root of unity. This example will also be used throughout the rest of this part of the thesis to illustrate various filter bank decompositions. A data word $u(z^{-1})$ of 10 symbols is encoded into a codeword $y(z^{-1})$ of 15 symbols by filtering with the generator polynomial

$$\begin{aligned} g_{\mathcal{R}}(z^{-1}) &= \prod_{k=3}^{7} (z^{-1} - \alpha^k) \\ &= \alpha^{10} + \alpha^9 z^{-1} + \alpha^{11} z^{-2} + \alpha^6 z^{-3} + \alpha^9 z^{-4} + \alpha^0 z^{-5}. \end{aligned}$$

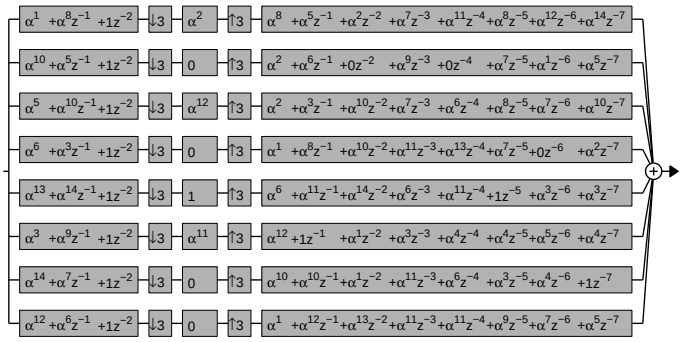


Figure 3.3: *overlap-add* filter bank ($L = 6, M = 8, N = 3$) for the $\mathcal{R}[15, 10]$ code described in Example 3.3. A dataword of 10 symbols padded with zeros is fed in the filter bank, resulting in a codeword of 15 symbols.

Since the roots of $g_{\mathcal{R}}(z^{-1})$ are $\nu - \kappa$ consecutive powers of α, the code $\mathcal{R}[15, 10]$ is an RS code with a Hamming distance $\delta = 6$, which is equal to the filter length L. Let us choose the downsampling factor $N = 3$; now, take $M = 8 \geq L + N - 1$. As roots of $r(z^{-1})$, we pick the following 8 elements of the Galois Field: $\alpha^8, \alpha^5, \alpha^{10}, \alpha^3, \alpha^{14}, \alpha^9, \alpha^7, \alpha^6$. The resulting *overlap-add* filter bank is shown in Figure 3.3. △

The OLA/OLS filter banks implementing RS codes have a few drawbacks. As can be seen by the matrix decomposition (see Figure 3.2), the generator matrix $\mathbf{G}_{\mathcal{R}}$ is factored into $\mathbf{G}_{\mathcal{C}}\mathbf{G}_{\mathcal{D}}\mathbf{G}_{\mathcal{A}}$, which can be seen as generator matrices of codes $\mathcal{C}, \mathcal{D}$ and $\mathcal{A}$ respectively. The code $\mathcal{A}$ adds more redundancy than required by the RS code rate $\frac{\kappa}{\nu}$. On the contrary, the code $\mathcal{D}$ has rate 1. Therefore, it does not add redundancy, while $\mathcal{C}$ removes the excess redundancy added by $\mathcal{A}$. This decomposition turns out to be not very useful for the applications discussed in part II. In fact, the reverse situation is desirable, in which all the redundancy is added by the subband codes only (code $\mathcal{D}$).

3.3 Filter Bank with Subcodes in Subbands

An OLA/OLS scheme can be easily extended to include (multiple tap) subband filters. Therefore, imagine that the coefficients $g[0], \ldots, g[L-1]$ of the polynomial $g(z^{-1})$ in Section 3.2 are themselves polynomials in z^{-N}, i.e.

$$g[l](z^{-N}) = \sum_{k=0}^{K-1} g[l, k] z^{-kN}. \tag{3.10}$$

The degree of these polynomials is given by K. Hence, a filter bank can implement a filter with length exceeding L if its coefficients $g[0], \ldots, g[L-1]$ are polynomials of degree K. Following the *Cook-Toom decomposition*, the diagonal elements of $\mathbf{D}$ will then also be functions of z^{-N}, eventually resulting in filters $d_0(z^{-1}), \ldots, d_{L-1}(z^{-1})$ with K coefficients in the subbands of the corresponding filter bank implementation. Relying on the superposition principle (see [78] for a similar proof), it can be shown that the filter bank implements a convolution with $g(z^{-1}) = z^{-N+1+\phi} \sum_{l=0}^{L-1} z^{-l} g[l](z^{-N})$. This can also be written in matrix notation as follows:

$$g(z^{-1}) = z^{-N+1+\phi} \begin{bmatrix} 1 & z^{-1} & z^{-2} & \ldots & z^{-(L-1)} \end{bmatrix} \cdot \underbrace{\begin{bmatrix} g[0,0] & g[0,1] & \ldots & g[0,K-1] \\ g[1,0] & g[1,1] & \ldots & g[1,K-1] \\ \vdots & \vdots & \ddots & \vdots \\ g[L-1,0] & g[L-1,1] & \ldots & g[L-1,K-1] \end{bmatrix}}_{\mathbf{\Gamma}} \cdot \begin{bmatrix} 1 \\ z^{-N} \\ z^{-2N} \\ \vdots \\ z^{-(K-1)N} \end{bmatrix} . \quad (3.11)$$

Note that $K = 1$ results in the filter bank described in Section 3.2. Note also that not every filter can be realized if $L < N$ (leading to missing powers of z^{-1} in Eq. 3.11), therefore we impose a second constraint: $L \geq N$.

Example 3.4 A realization of $\mathcal{R}[15, 10]$ is obtained 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 roots of $r(z^{-1})$ are chosen $0, 1, \alpha, \alpha^4, \alpha^2$. Since the filter $g_{\mathcal{R}}(z^{-1})$ has 6 coefficients, the matrix $\mathbf{\Gamma}$ is 3×2 ($K = 2$) as shown below ($\phi = 0$).

$$g_{\mathcal{R}}(z^{-1}) = z^{-2} \begin{bmatrix} 1 & z^{-1} & z^{-2} \end{bmatrix} \underbrace{\begin{bmatrix} \alpha^{10} & \alpha^6 \\ \alpha^9 & \alpha^9 \\ \alpha^{11} & 1 \end{bmatrix}}_{\mathbf{\Gamma}} \begin{bmatrix} 1 \\ z^{-3} \end{bmatrix} . \quad (3.12)$$

The resulting filter bank implementation can be found in Figure 3.5 and generates the same code $\mathcal{R}[15, 10]$ as in Example 3.3. The corresponding matrix factorization is shown in Figure 3.4. △

Remark that if $L = N$, which is the case in the previous example, this filter bank can be found from the polyphase components of $\mathbf{u}$, $\mathbf{g}$ *and* $\mathbf{y}$ *by applying an aperiodic convolution algorithm. This special case is described by Vetterli in [78].*

If the M-th roots of unity are chosen as interpolation points (roots of $r(z^{-1})$), the resulting filter bank (in $\mathbb{C}$) is referred to as the *Short Time Fourier Transform (STFT) bank*. This second special case is also studied by Vetterli [78]. Its

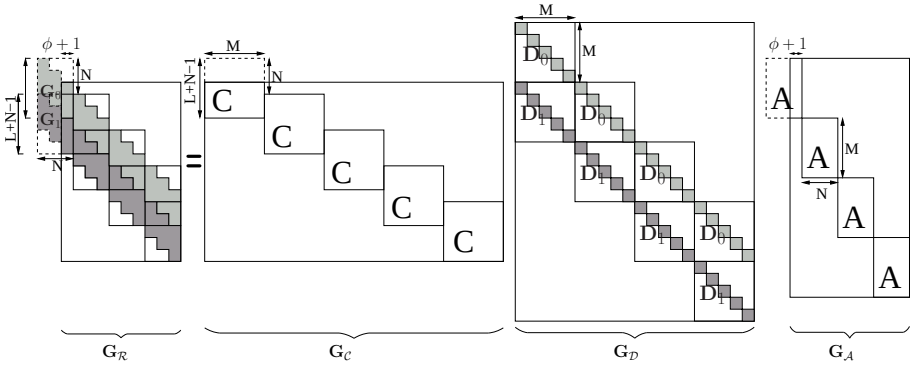


Figure 3.4: Matrix decomposition for the filter bank in Example 3.4 implementing $\mathcal{R}\,[15, 10]$. The parameters are $K = 2$, $L = N = 3$, $M = 5$, $\phi = 0$.

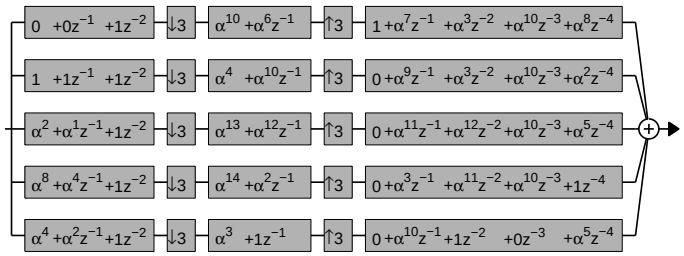


Figure 3.5: Filter bank with subcodes in each subband for $\mathcal{R}\,[15, 10]$ (Example 3.4)

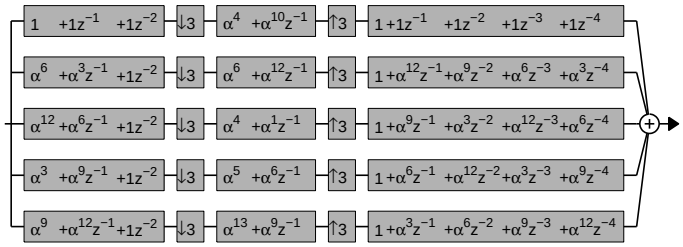


Figure 3.6: STFT-like filter bank based on the *Mattson-Solomon Polynomial* for the RS code $\mathcal{R}\,[15, 10]$

Galois field counterpart does also exist (See Section 2.1.1: if M divides ν, e.g. if $M = 5$, a multiplicative subgroup $\alpha^0 = 1$, α^3, α^6, α^9, α^{12}, ($\alpha^{15} = 1$) exists.) Using these elements as the roots of $r(z^{-1})$, we obtain the filter bank shown in Figure 3.6. As mentioned before, in a coding theory context, the analysis and synthesis bank calculate the *Mattson-Solomon Polynomial* (See [67, Ch. 8, §6].) of their inputs, also known as the DFT in the GF. As will be seen further, this property is very important. The filter bank in Figure 3.6 can thus be seen as *the (GF) counterpart of the Short Time Fourier Transform (STFT) bank*. In the complex field, the STFT bank was introduced by [16, 56].

In this section, one of the problems presented at the end of Section 3.2 was solved: the subband filters (terminated convolutions) indeed add redundancy now. However, the problem of the synthesis and analysis bank codes ($\mathcal{C}$ and $\mathcal{A}$, which are hard to decode) also adding redundancy does remain. In the next section, this problem will be tackled by developing a critically subsampled filter bank.

3.4 Critically Subsampled Filter Bank

In this section, a critically subsampled filter bank implementation is built for the family of RS codes with $\mathcal{C}$ and $\mathcal{A}$ rate 1 codes. In order to achieve this goal, the constraint $M \geq L+N-1$ must be violated ($M = N$), resulting in a so-called *critically subsampled filter bank* that does *not* implement a convolution with $g_{\mathcal{R}}(z^{-1})$ but a convolution modulo $r(z^{-1})$. For reasons that will become clear, we choose $r(z^{-1}) = 1 + z^{-M}$ which effectively leads to a circular convolution. A decomposition of $r(z^{-1})$ in first degree polynomials only exists if M divides $\nu = q - 1$. Let $\nu' = \frac{\nu}{M}$. According to Fermat's little theorem (Theorem 2.1), a multiplicative subgroup generated by $\gamma = \alpha^{\nu'}$ can be found and

$$r(z^{-1}) = 1 + z^{-M} = \prod_{k=0}^{M-1} (z^{-1} + \gamma^k). \tag{3.13}$$

Hence, the resulting filter bank is again based on the DFT in the Galois field (See Eq. 3.7).

Essentially, a filter bank with a large enough number of bands ($M \geq L + N - 1$, i.e. an oversampled filter bank) splits one large Toeplitz matrix into many smaller submatrices, which are Toeplitz matrices themselves and hence can be decomposed according to Cook-Toom's algorithm. However, if this condition is not met, the submatrices show overlap and are not Toeplitz any more (convolution modulo $r(z^{-1})$). The latter effect is often referred to as *aliasing* due to insufficient oversampling. In the limit of a critically subsampled filter bank ($M = N$), these submatrices become circulant (at least when based on the *Mattson-Solomon Polynomial*) and do not overlap. The ideas developed in this section show that the inherent cyclic character of the RS code maps very well onto the aliased behavior of a critically subsampled filter bank.

In Section 3.4.1, a generator matrix of a code equivalent to $\mathcal{R}$ is constructed. The generator matrix $\mathbf{G}_{\mathcal{R}}^{V}$ has the nice property that interleaving its rows results in a block Toeplitz matrix with circulant blocks. The interleaved code, referred to as $\mathcal{R}^{\Pi_{\nu'}}[\nu, \kappa]$ in Section 3.4.2, is also equivalent to the RS code and inherits its distance properties. Finally, the result is further simplified in Section 3.4.3.

The techniques presented in this chapter can be generally applied to any RS code as long as ν and κ are not coprime. The number of bands M should not only divide ν, but also κ such that $\nu' = \frac{\nu}{M}$ and $\kappa' = \frac{\kappa}{M}$. These constraints are necessary for the block Toeplitz matrix to exist. However, the following chapter shows that the latter constraint on κ can be removed. This requires a different approach to the problem which gives less insight in the relation between filter banks and RS codes. For now, we assume that ν and κ are not coprime.

3.4.1 Step 1: Equivalent RS Code generated by $\mathbf{G}_{\mathcal{R}}^{V}$

Recall the following 3 properties of RS codes:

- RS codes have a minimum distance $\delta = \nu - \kappa + 1$, therefore a unique codeword (up to a scalar multiplication) can always be found with nonzeros on δ arbitrarily chosen positions. This can easily be seen as follows: Imagine a codeword $\mathbf{y}$ must be found which is nonzero at the positions $\mathbf{a} = \begin{bmatrix} a[0] & a[1] & \dots & a[\delta - 1] \end{bmatrix}$. Hence, $\mathbf{y}$ can be found as a solution of the following system:
$$\mathbf{H}_{\mathcal{R}}[:, \mathbf{a}]\, \mathbf{y}[\mathbf{a}] = \mathbf{0}\,. \tag{3.14}$$
The submatrix of the parity check matrix $\mathbf{H}_{\mathcal{R}}$ (dimensions $\nu - \kappa = \delta - 1 \times \delta$) has full column rank $\delta - 1$ (according to the BCH bound in Theorem 2.10). Hence, there is a unique nonzero solution for $\mathbf{y}$ (up to a scalar).

- RS codes are cyclic codes which means that every cyclic shift of a codeword is also a codeword:

$$y(z^{-1}) \in \mathcal{R}[\nu,\kappa] \Rightarrow z^{-k}y(z^{-1}) \bmod (1-z^{-\nu}) \in \mathcal{R}[\nu,\kappa], \forall k\,. \quad (3.15)$$

- RS codes are linear codes; therefore, the freedom to choose the codewords in the generator matrix $\mathbf{G}_{\mathcal{R}}$ may be exploited: κ linearly independent codewords of the original code can serve as columns of a (new) generator matrix, which generates an equivalent code.

These properties are subsequently used to construct an equivalent code generated by $\mathbf{G}^V_{\mathcal{R}}$ as follows:

- Using the first property, an $M \times K$ matrix $\mathbf{\Gamma}$ with $M = L = N$ is determined in such a way that $y(z^{-1})$ corresponds to a codeword of $\mathcal{R}[\nu,\kappa]$.

$$y(z^{-1}) = \left[z^{-(0:L-1)\nu'}\right] \mathbf{\Gamma} \left[z^{-(0:K-1)}\right]^T \bmod (1-z^{-\nu})\,.$$

  Therefore, only $MK - \delta + 1$ coordinates can be chosen zero, while the others can be uniquely determined (up to a scalar) such that $y(z^{-1}) \in \mathcal{R}[\nu,\kappa]$. This equation reminds us of Eq. 3.11, however note that the powers of z^{-1} differ. In Section 3.4.2 below, it is seen that after a simple interleaving operation this matrix takes the role of $\mathbf{\Gamma}$ as defined in Eq. 3.11.

- According to the second property, every shifted version $z^{-k-l\nu'}y(z^{-1}) \bmod (1-z^{-\nu})$, for $k = 0 : K-1$ and $l = 0 : L-1$ is also a codeword and belongs to $\mathcal{R}[\nu,\kappa]$:

$$\left[z^{-(l:l+L-1)\nu'}\right] \mathbf{\Gamma} \left[z^{-(k:k+K-1)}\right]^T \bmod (1-z^{-\nu}) \in \mathcal{R}[\nu,\kappa]\,.$$

- Corresponding to the third property, these shifted versions of $y(z^{-1})$ can serve as columns of $\mathbf{G}^V_{\mathcal{R}}$, a specially structured generator matrix of an *equivalent* RS code.

This code contains the same codewords as $\mathcal{R}[\nu,\kappa]$, but the mapping between a specific dataword and its codeword has changed corresponding to a basis transformation matrix $\mathbf{V}$. Therefore, the following relation holds: $\mathbf{G}_{\mathcal{R}} = \mathbf{G}^V_{\mathcal{R}}\mathbf{V}$.

Example 3.5 For $\mathcal{R}[15,10]$, let $K = 2$ and $L = M = N = 5$ (dividing $\nu = 15$). We choose the last 2 rows of $\mathbf{\Gamma}$ to be 0, and $\Gamma[0,0]$ to be $\alpha^0 = 1$. The remaining five elements are calculated so that $y(z^{-1})$ is in the subspace generated by $\mathbf{G}_{\mathcal{R}}$.

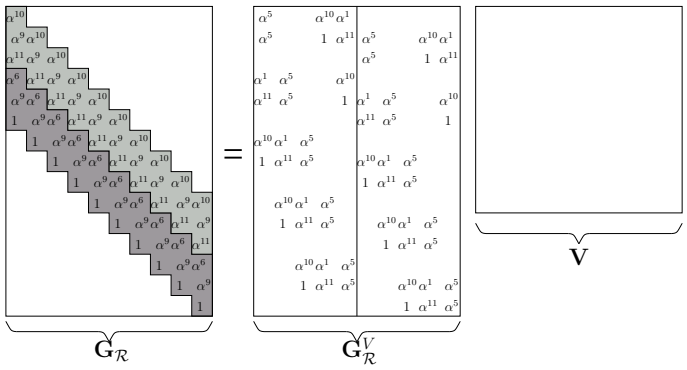


Figure 3.7: Step 1: constructing an equivalent RS code for $\mathcal{R}\,[15, 10]$.

This gives the unique codeword

$$y(z^{-1}) = \begin{bmatrix} z^{-0} & z^{-3} & z^{-6} & z^{-9} & z^{-12} \end{bmatrix} \underbrace{\begin{bmatrix} 1 & \alpha^5 \\ \alpha^{11} & \alpha^{11} \\ \alpha^5 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}}_{\mathbf{\Gamma}} \begin{bmatrix} z^{-0} \\ z^{-1} \end{bmatrix}. \tag{3.16}$$

Note that every shifted version $z^{-k-l\nu'}$, for $k = 0:1$ and $l = 0:4$ is also a codeword and belongs to $\mathcal{R}\,[15, 10]$. Stacking the codewords yields the matrix $\mathbf{G}^{V}_{\mathcal{R}}$, shown in Figure 3.7. △

3.4.2 Step 2: Equivalent RS Code generated by $\mathbf{G}^{V}_{\mathcal{R}^{\Pi_{\nu'}}}$

In the next step, the codeword $y^{\Pi_{\nu'}}(z^{-1})$ is constructed by stacking the ν' polyphase components of $y(z^{-1})$. Defining the k-th polyphase component of $y(z^{-1})$ as

$$y_k^{[\nu']}(z^{-1}) = \sum_{m=0}^{M-1} y\,[k + m\nu']\, z^{-k}\,, \tag{3.17}$$

$y^{\Pi_{\nu'}}(z^{-1})$ is defined as follows:

$$y^{\Pi_{\nu'}}(z^{-1}) = \sum_{k=0}^{M-1} y_k^{[\nu']}(z^{-1}) z^{-Mk}\,. \tag{3.18}$$

This reordering defines an interleaver matrix $\mathbf{\Pi}_{\nu'}$ such that $\mathbf{\Pi}_{\nu'}\mathbf{y}^{\mathbf{\Pi}}_{\nu'} = \mathbf{y}$. In general, an interleaver matrix

$$\mathbf{\Pi}_x = \begin{bmatrix} \mathbf{I}_{\frac{\nu}{x}} \otimes \mathbf{I}_x\,[:, 0] & \mathbf{I}_{\frac{\nu}{x}} \otimes \mathbf{I}_x\,[:, 1] & \ldots & \mathbf{I}_{\frac{\nu}{x}} \otimes \mathbf{I}_x\,[:, x-1] \end{bmatrix}, \tag{3.19}$$

where $\mathbf{I}_x$ is the $x \times x$ identity matrix. Applying the interleaving operation to $\mathbf{G}_{\mathcal{R}}^V$ results in a $\mathbf{\Pi}_{\nu'}\mathbf{G}_{\mathcal{R}^{\Pi_{\nu'}}}^V = \mathbf{G}_{\mathcal{R}}^V$, leading to the following decomposition of the original (Toeplitz) generator matrix:

$$\boxed{\mathbf{G}_{\mathcal{R}} = \mathbf{\Pi}_{\nu'} \underbrace{\mathbf{G}_{\mathcal{C}}\mathbf{G}_{\mathcal{D}}\mathbf{G}_{\mathcal{A}}}_{\mathbf{G}_{\mathcal{R}^{\Pi_{\nu'}}}^V} \mathbf{V}} \tag{3.20}$$

$\mathbf{G}_{\mathcal{R}^{\Pi_{\nu'}}}^V$ is then a *block Toeplitz generator matrix with circulant blocks*. This shows that with the interleaving, a circulant shift modulo ν of the codewords is translated into circular shifts modulo M, yielding small circulant blocks. Since interleaving the generator matrix interleaves the whole codebook, the distance properties of the RS code are preserved. Hence, this (new) code is an RS code too, and will be denoted with $\mathcal{R}^{\Pi_{\nu'}}[\nu, \kappa]$. The generator matrix of this code can be found in Figure 3.8. More formally, $\mathcal{R}^{\Pi_{\nu'}}[\nu, \kappa]$ is defined as follows:

$$\mathcal{R}^{\Pi_{\nu'}}[\nu, \kappa] = \{\mathbf{y} | \mathbf{\Pi}_{\nu'}\mathbf{y} \in \mathcal{R}[\nu, \kappa]\}.$$

The interleaved codeword $y^{\Pi_{\nu'}}(z^{-1})$ shows the following structure:

$$y^{\Pi_{\nu'}}(z^{-1}) = \left[z^{-(0:L-1)}\right] \mathbf{\Gamma} \left[z^{-(0:K-1)N}\right]^T. \tag{3.21}$$

Note that $y^{\Pi_{\nu'}}(z^{-1})$ has the correct form to apply the *Cook-Toom's algorithm*: The matrix to the left of $\mathbf{\Gamma}$ has ascending powers of z^{-1} while the matrix to the right shows ascending powers of z^{-N} (as in Eq. 3.11). Since $r(z^{-1}) = 1 + z^{-M}$ (Eq. 3.13), applying the *Cook-Toom's algorithm* with $L = M = N$ for each k-th column of $\mathbf{\Gamma}$, $k = 0 : K-1$ exactly implements the circulant block matrices $\mathbf{G}_k, k = 0 : K-1$, found in $\mathbf{G}_{\mathcal{R}^{\Pi_{\nu'}}}^V$. To build the filter bank, ϕ must be fixed to $(-N+1) \bmod N$ such that the first block in the generator matrix $\mathbf{G}_{\mathcal{R}^{\Pi_{\nu'}}}^V$ is complete.

Example 3.6 For our example, the interleaved codeword is

$$y^{\Pi_{\nu'}}(z^{-1}) = \begin{bmatrix} z^{-0} & z^{-1} & z^{-2} & z^{-3} & z^{-4} \end{bmatrix} \underbrace{\begin{bmatrix} 1 & \alpha^5 \\ \alpha^{11} & \alpha^{11} \\ \alpha^5 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}}_{\mathbf{\Gamma}} \begin{bmatrix} z^{-0} \\ z^{-5} \end{bmatrix} \tag{3.22}$$

Shifting this codeword as described by Eq. 3.4.1 results in the $\mathbf{G}_{\mathcal{R}^{\Pi_{\nu'}}}^V$ of Figure 3.9. The corresponding filter bank is shown in Figure 3.10. The 2 decompositions based on *Cook-Toom's algorithm* used to construct this filter bank

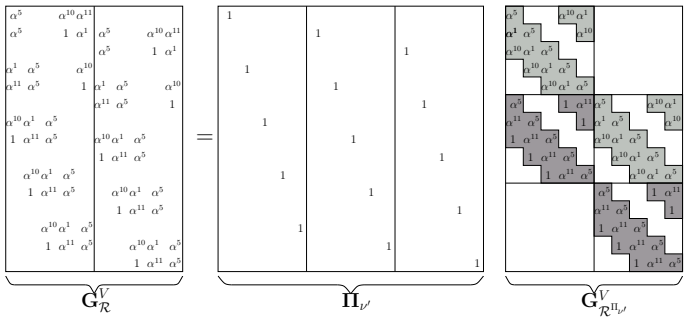


Figure 3.8: Step 2: Constructing the code $\mathcal{R}^{\Pi_{\nu'}}[15, 10]$ by interleaving $\mathbf{G}_{\mathcal{R}^V}(15, 10)$, yielding a block-Toeplitz matrix with circulant blocks.

are shown below:

$$\underbrace{\begin{bmatrix} \alpha^5 & & & \alpha^{10} & \alpha^1 \\ \alpha^1 & \alpha^5 & & & \alpha^{10} \\ \alpha^{10} & \alpha^1 & \alpha^5 & & \\ & \alpha^{10} & \alpha^1 & \alpha^5 & \\ & & \alpha^{10} & \alpha^1 & \alpha^5 \end{bmatrix}}_{\mathbf{G}_0} = \underbrace{\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & \alpha^{12} & \alpha^9 & \alpha^6 & \alpha^3 \\ 1 & \alpha^9 & \alpha^3 & \alpha^{12} & \alpha^6 \\ 1 & \alpha^6 & \alpha^{12} & \alpha^3 & \alpha^9 \\ 1 & \alpha^3 & \alpha^6 & \alpha^9 & \alpha^{12} \end{bmatrix}}_{\mathbf{F}_5^{-1}} \underbrace{\begin{bmatrix} \alpha^4 & & & & \\ & \alpha^{10} & & & \\ & & \alpha^5 & & \\ & & & \alpha^6 & \\ & & & & \alpha^3 \end{bmatrix}}_{\mathbf{D}_0} \underbrace{\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & \alpha^3 & \alpha^6 & \alpha^9 & \alpha^{12} \\ 1 & \alpha^6 & \alpha^{12} & \alpha^3 & \alpha^9 \\ 1 & \alpha^9 & \alpha^3 & \alpha^{12} & \alpha^6 \\ 1 & \alpha^{12} & \alpha^9 & \alpha^6 & \alpha^3 \end{bmatrix}}_{\mathbf{F}_5}$$

$$\underbrace{\begin{bmatrix} \alpha^5 & & & 1 & \alpha^{11} \\ \alpha^{11} & \alpha^5 & & & 1 \\ 1 & \alpha^{11} & \alpha^5 & & \\ & 1 & \alpha^{11} & \alpha^5 & \\ & & 1 & \alpha^{11} & \alpha^5 \end{bmatrix}}_{\mathbf{G}_1} = \underbrace{\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & \alpha^{12} & \alpha^9 & \alpha^6 & \alpha^3 \\ 1 & \alpha^9 & \alpha^3 & \alpha^{12} & \alpha^6 \\ 1 & \alpha^6 & \alpha^{12} & \alpha^3 & \alpha^9 \\ 1 & \alpha^3 & \alpha^6 & \alpha^9 & \alpha^{12} \end{bmatrix}}_{\mathbf{F}_5^{-1}} \underbrace{\begin{bmatrix} \alpha^{14} & & & & \\ & \alpha^4 & & & \\ & & \alpha^{13} & & \\ & & & \alpha^3 & \\ & & & & \alpha^{14} \end{bmatrix}}_{\mathbf{D}_1} \underbrace{\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & \alpha^3 & \alpha^6 & \alpha^9 & \alpha^{12} \\ 1 & \alpha^6 & \alpha^{12} & \alpha^3 & \alpha^9 \\ 1 & \alpha^9 & \alpha^3 & \alpha^{12} & \alpha^6 \\ 1 & \alpha^{12} & \alpha^9 & \alpha^6 & \alpha^3 \end{bmatrix}}_{\mathbf{F}_5}$$

$$\mathbf{D}_0 = \text{Diag}\left(\mathbf{F}_5 \mathbf{\Gamma}[:, 0]\right)$$
$$\mathbf{D}_1 = \text{Diag}\left(\mathbf{F}_5 \mathbf{\Gamma}[:, 1]\right)$$

△

Note that Solomon discovered a circulant structure in the RS codes [68], however this structure differs from the block Toeplitz structure with circulant blocks derived here which is indeed a filter bank-like structure.

As a second remark, note that the *overlap-add* variant equals the *overlap-save* filter bank, if $\delta = 1$ in *Cook-Toom's algorithm*. Then, the DFT matrix $\mathbf{F}_M$ of the analysis bank is circularly shifted one step to the right resulting in $\mathbf{C} = \mathbf{J}\mathbf{A}^T$ (See also Figure 3.2). Signal flow transposition on this filter bank gives us exactly the same filter bank. Note that in general, for the *overlap-add* bank, the overlap occurs in the synthesis bank, where the filters have more coefficients

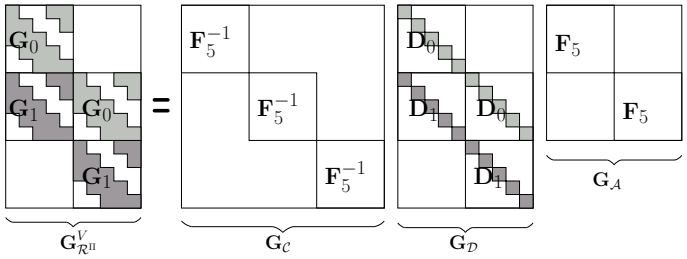


Figure 3.9: Step 2: Matrix decomposition for the critically downsampled ($L = M = N = 5, \phi = 4$) filter bank for $\mathcal{R}^{\Pi_{\nu'}}[15, 10]$ code, found in Example 3.5.

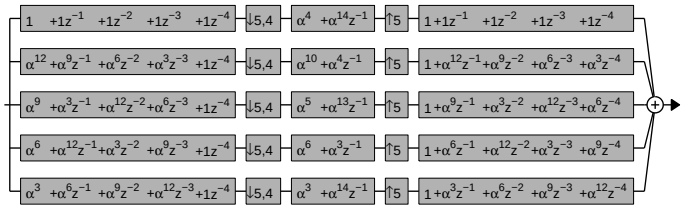


Figure 3.10: Critically subsampled ($L = M = N = 5, \phi = 4$) filter bank for $\mathcal{R}[15, 10]$ code, found in Example 3.5.

(M) than the analysis filters (N). This is the other way around for the *overlap-save* variant. When M is gradually decreased, the overlap diminishes and vanishes eventually when the filter bank becomes critically subsampled. Recall that this filter bank does no longer implement a filter, but can still implement an RS code.

3.4.3 Step 3: Equivalent RS Code generated by $\mathbf{G}^P_{\mathcal{R}^{\Pi_{\nu'}}}$

In the last step, the RS code can be further simplified since $\mathbf{G}_{\mathcal{A}}$ does not add redundancy any more. It can be seen as an extra mapping on the dataword before it is encoded by

$$\mathbf{G}^P_{\mathcal{R}^{\Pi_{\nu'}}} = \mathbf{G}_{\mathcal{C}}\mathbf{G}_{\mathcal{D}}\,.$$

Therefore, with $\mathbf{P} = \mathbf{G}_{\mathcal{A}}\mathbf{V}$, the original generator matrix is decomposed as

$$\mathbf{G}_{\mathcal{R}} = \mathbf{\Pi}_{\nu'}\mathbf{G}^P_{\mathcal{R}^{\Pi_{\nu'}}}\mathbf{P}\,.$$

Example 3.7 The filter bank implementing $\mathcal{R}^{\Pi_{\nu'}}[15, 10]$ (generated by $\mathbf{G}^P_{\mathcal{R}^{\Pi_{\nu'}}}$) is shown in Figure 3.11. △

Now, our goal is achieved; the filter bank is critically subsampled. This property allows us to build a SISO decoder, as will be discussed in Part II.

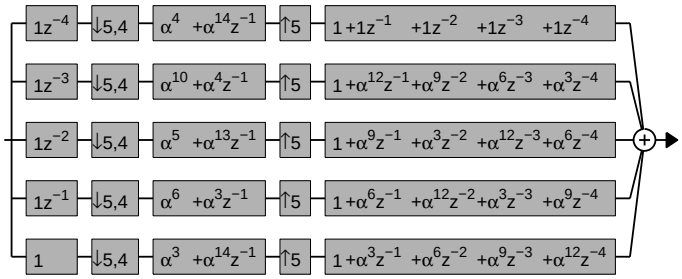


Figure 3.11: Critically downsampled filter bank implementing $\mathcal{R}^{\Pi_{\nu'}}$ [15, 10]. The downsample block $\downarrow N, \phi$ denotes that the sequence is first shifted by ϕ (multiplied by z^{ϕ}) before the actual downsampling (starting with the first sample) takes place.

The synthesis bank does not add redundancy, it links the subband codes together in a complex way such that the overall code is an RS code. Finally, the subband filters are – also literally speaking – the core codes: all redundancy is added by M terminated convolutional codes in the subbands that have the same rate as the overall code ($\rho = 10/15 = 2/3$ in our example). In a SISO decoder, these codes can be efficiently decoded using the BCJR algorithm.

In the next section, the existence of a critically subsampled filter bank that is systematic itself is further developed. These results are more of theoretical interest.

3.5 Critically Subsampled IIR Filter Bank for Systematic RS?

In this section, it is seen that a systematic critically subsampled IIR filter bank can easily be derived from its FIR counterpart. Therefore, let us recall the key Eq. 3.20:

$$\mathbf{G}_{\mathcal{R}} = \mathbf{\Pi}_{\nu'} \mathbf{G}_{\mathcal{C}} \mathbf{G}_{\mathcal{D}} \mathbf{G}_{\mathcal{A}} \mathbf{V} \,. \tag{3.23}$$

In this equation, $\mathbf{G}_{\mathcal{C}} \mathbf{G}_{\mathcal{D}} \mathbf{G}_{\mathcal{A}}$ represents the filter bank. In a first step, the generator matrix $\mathbf{G}_{\mathcal{D}}$ of $\mathcal{D}$, consisting of all the subband filters, is transformed into its systematic form:

$$\mathbf{G}_{\mathcal{D}^{sys}} = \mathbf{G}_{\mathcal{D}} \mathbf{P}^{-1} \text{ with } \mathbf{P} = \mathbf{G}_{\mathcal{D}} \left[0 : \kappa - 1, 0 : \kappa - 1 \right] . \tag{3.24}$$

In the second step, define $\mathbf{P}'$ such that

$$\mathbf{G}_{\mathcal{A}} \mathbf{P}' = \mathbf{P} \mathbf{G}_{\mathcal{A}} \,. \tag{3.25}$$

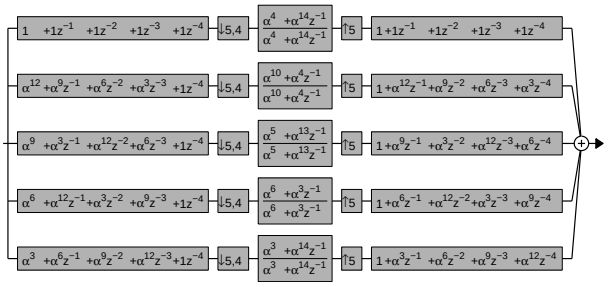


Figure 3.12: Critically subsampled IIR (systematic) filter bank ($M = 5$) for $\mathcal{R}[15, 10]$.

This is only possible since $\mathbf{P}$ is a block matrix with diagonal $M \times M$ blocks and $\mathbf{G}_{\mathcal{A}}$ is a block matrix composed of DFT matrices:

$$\mathbf{G}_{\mathcal{A}} = (\mathbf{I}_{\kappa'} \otimes \mathbf{F}_M) \ . \tag{3.26}$$

Substituting Equations 3.24, 3.25 in Eq. 3.23 gives:

$$\begin{aligned} \mathbf{G}_{\mathcal{R}} &= \mathbf{\Pi}_{\nu'} \mathbf{G}_{\mathcal{C}} \mathbf{G}_{\mathcal{D}}^{sys} \mathbf{P} \mathbf{G}_{\mathcal{A}} \mathbf{V} && (3.27) \\ &= \mathbf{\Pi}_{\nu'} \underbrace{\mathbf{G}_{\mathcal{C}} \mathbf{G}_{\mathcal{D}}^{sys} \mathbf{G}_{\mathcal{A}}}_{\mathbf{G}_{\mathcal{R}^{\Pi_{\nu'}}}^{P'V}} \mathbf{P}' \mathbf{V} && (3.28) \end{aligned}$$

It is clear that the code defined by the generator matrix $\mathbf{G}_{\mathcal{R}^{\Pi_{\nu'}}}^{P'V}$ is equivalent to the RS code $\mathcal{R}[\nu, \kappa]$. The block-structure of this generator matrix is

$$\mathbf{G}_{\mathcal{R}^{\Pi_{\nu'}}}^{PV'} = \begin{bmatrix} \mathbf{I}_M & & & \\ & \mathbf{I}_M & & \\ & & \ddots & \\ & & & \mathbf{I}_M \\ \mathbf{C}_{\kappa',0} & \mathbf{C}_{\kappa',1} & \dots & \mathbf{C}_{\kappa',\kappa'-1} \\ \mathbf{C}_{\kappa'+1,0} & \mathbf{C}_{\kappa'+1,1} & \dots & \mathbf{C}_{\kappa'+1,\kappa'-1} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{C}_{\nu'-1,0} & \mathbf{C}_{\nu'-1,1} & \dots & \mathbf{C}_{\nu'-1,\kappa'-1} \end{bmatrix} . \tag{3.29}$$

Each $M \times M$ matrix $\mathbf{C}_{i,j}$ is a circulant matrix. Note that due to its structure, κ and M cannot be coprime. Following the same reasoning as in Section 4.3.2, it is clear that this block structure can be implemented by an IIR filter bank. The corresponding filter bank is critically subsampled and the M subband filters are terminated recursive convolutional codes, which are systematic themselves. Figure 3.12 shows the systematic IIR filter bank implementation of the code $\mathcal{R}[15, 10]$.

3.6 Conclusion

This chapter focuses on critically subsampled filter bank decompositions of non-systematic RS codes, that break RS codes into convolutional subcodes. In a first step, *overlap-add* and *overlap-save* filter banks that operate in the GF are derived, based on *Cook-Toom's algorithm*. These filter banks have zero order tap filters in their subbands. To allow error correction, in a second step, a filter bank is built that has nonzero 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 subsampled. In a third step, it is explained how a critically subsampled filter bank can be obtained. It is again based on the *Mattson-Solomon polynomial* or DFT in the GF and on the fact that it can be used to diagonalize a circulant matrix. The existence of a filter bank representation is remarkable noting that it can never perform linear filtering operations. It is seen that the aliasing – due to the critical subsampling – corresponds with the cyclic character of the RS code. The techniques presented here can be generally applied to any RS code as long as ν and κ are not coprime. Finally, a critically subsampled filter bank implementing a systematic version is presented. The latter can be simply obtained if the subband codes are rewritten in their systematic form.

Chapter 4

Reed-Solomon Codes and Filter Banks: An Algebraic Approach

4.1 Introduction

If the block length ν and dimension κ of the RS code are coprime, or e.g. if a filter bank with fewer bands is desired, the methods described in the previous chapter fail to produce a suitable decomposition. In this chapter, all constraints on the dimension κ of the RS code are removed, such that there is flexibility in the choice of the code rate. First, the reverse Cook-Toom decomposition is discussed, as it is an important tool used in the rest of the chapter. Next, Section 4.3 shows that critically subsampled filter banks for RS codes with M bands exist as long as M divides ν (denoted as $M|\nu$). In the case of RS codes, $\nu = q - 1$ and thus M *should divide* $q - 1$, such that a multiplicative subgroup in $\mathbb{F}_q$ can be identified with M elements. This condition is crucial for an M-point DFT to exist in a Galois field (see Section 2.1.1). As a first contribution, the filter bank structure is studied from a purely algebraic point of view. Such a representation is more suitable for building a filter bank decomposition for a more general collection of RS codes, thereby indeed eliminating the non-coprimeness constraint. This is an important step, since in many practical codes, the codeword and dataword length are coprime, e.g. the popular $\mathcal{R}[255, 223]$. The theory will be illustrated by constructing a filter bank with $M = 3$ instead of 5 bands for the example of the $\mathcal{R}[15, 10]$ code of the previous chapter.

Finally, in Section 4.4, the focus shifts towards the more general family of BCH codes. For these codes, ν can differ from $q-1$ such that $M|\nu$ does no longer guarantee the existence of an M-point DFT transform. Filter bank representations however do still exist, as will be shown.

4.2 Reverse Cook-Toom Decomposition

In Section 3.2, Cook-Toom's algorithm was presented. In this section, its reverse is presented such that a given $M \times M$ diagonal matrix $\mathbf{D}$ can be decomposed into a product of 3 structured matrices. The middle factor is a Toeplitz matrix $\mathbf{G}$ implementing a filter of length L. The reverse Cook-Toom algorithm will be crucial later on.

Theorem 4.1 *Any $M \times M$ diagonal matrix $\mathbf{D}$ with $L-1$ zeros on its diagonal can be decomposed as a product $\mathbf{D} = \tilde{\mathbf{A}}\mathbf{G}\mathbf{W}$. In this decomposition, $\tilde{\mathbf{A}}$ is an $M \times M$ Vandermonde matrix, $\mathbf{G}$ is an $M \times N$ ($N = M-L+1$) Toeplitz matrix implementing a filter $g(z^{-1})$ of length L and $\mathbf{W}$ is an $N \times M$ matrix.*

Proof: Choosing M distinct roots $\boldsymbol{\rho} = \begin{bmatrix}\rho[0] & \rho[1] & \dots \rho[M-1]\end{bmatrix}$, let $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ be the Cook-Toom matrices corresponding to these roots (Theorem 3.1). Let $\boldsymbol{\tau}$ be a vector containing the non-zero positions of $\mathbf{D}$, and $\overline{\boldsymbol{\tau}}$ the complementary set of zero positions. Then $\mathbf{G}$ can be easily found since the roots of $g(z^{-1})$ are $\rho[\overline{\boldsymbol{\tau}}]$. $\mathbf{G}$ can now be written as

$$\mathbf{G} = \mathbf{C}\mathbf{D}'\mathbf{A} \tag{4.1}$$

with

$$\mathbf{D}' = \mathbf{B}\,\mathrm{Null}\left(\mathbf{B}\left[\overline{\boldsymbol{\tau}}, :\right]\right). \tag{4.2}$$

Note that $\mathbf{D}'$ and $\mathbf{D}$ are zero at the same positions, but the other elements (can) differ. To calculate the $M-L+1 \times M$ matrix $\mathbf{W}$, let $\underline{\mathbf{T}}$ be the square submatrix of $\mathbf{T} = \mathbf{C}^{-1}\mathbf{G}$ with the zero rows omitted: $\underline{\mathbf{T}} = \mathbf{T}[\boldsymbol{\tau}, :]$. Correspondingly, define the square matrices $\underline{\mathbf{D}} = \mathbf{D}[\boldsymbol{\tau}, \boldsymbol{\tau}]$, $\underline{\mathbf{D}}' = \mathbf{D}'[\boldsymbol{\tau}, \boldsymbol{\tau}]$ and $\underline{\mathbf{A}} = \mathbf{A}[\boldsymbol{\tau}, :]$. The matrix $\underline{\mathbf{T}}$ can now be written as:

$$\begin{aligned} \underline{\mathbf{T}} &= \underline{\mathbf{D}}'\underline{\mathbf{A}} && (4.3)\\ &= \underline{\mathbf{D}}\underbrace{\underline{\mathbf{D}}^{-1}\underline{\mathbf{D}}'}_{\underline{\mathbf{W}}^{-1}}\underline{\mathbf{A}}\,. && (4.4) \end{aligned}$$

Inserting the zeros back in $\underline{\mathbf{T}}$, $\underline{\mathbf{D}}$ and $\underline{\mathbf{W}}$ ($\mathbf{W}[:, \boldsymbol{\tau}] = \underline{\mathbf{W}}$) gives

$$\mathbf{D} = \mathbf{T}\mathbf{W} = \mathbf{C}^{-1}\mathbf{G}\mathbf{W}\,. \tag{4.5}$$

As shown at the end of Section 3.2, $\tilde{\mathbf{A}} = \mathbf{C}^{-1}$ is a Vandermonde matrix (Eq. 3.5) which completes the proof. □

Example 4.1 To illustrate the reverse Cook-Toom algorithm, consider a 5×5 ($M = 5$) arbitrary diagonal matrix $\mathbf{D}$ with a zero in the second diagonal position ($L = 2$): $\mathbf{D} = \mathrm{Diag}\left(\left[\alpha^2\, 0\, \alpha^2\, 1\, 1\right]\right)$. Choosing Cook-Toom's roots $\left[\alpha^2\, \alpha^5\, \alpha^8\, \alpha^{11}\, \alpha^{14}\right]$, the following decomposition is found using the construction given in the proof:

$$\underbrace{\begin{bmatrix} \alpha^2 & & & & \\ & 0 & & & \\ & & \alpha^2 & & \\ & & & 1 & \\ & & & & 1 \end{bmatrix}}_{\mathbf{D}} = \underbrace{\begin{bmatrix} 1 & \alpha^2 & \alpha^4 & \alpha^6 & \alpha^8 \\ 1 & \alpha^5 & \alpha^{10} & 1 & \alpha^5 \\ 1 & \alpha^8 & \alpha^1 & \alpha^9 & \alpha^2 \\ 1 & \alpha^{11} & \alpha^7 & \alpha^3 & \alpha^{14} \\ 1 & \alpha^{14} & \alpha^{13} & \alpha^{12} & \alpha^{11} \end{bmatrix}}_{\tilde{\mathbf{A}}} \underbrace{\begin{bmatrix} \alpha^5 & & & & \\ 1 & \alpha^5 & & & \\ & 1 & \alpha^5 & & \\ & & 1 & \alpha^5 & \\ & & & 1 & \end{bmatrix}}_{\mathbf{G}} \underbrace{\begin{bmatrix} \alpha^{12} & 0 & \alpha^{12} & \alpha^{10} & \alpha^{10} \\ \alpha^6 & 0 & \alpha^3 & \alpha^{12} & \alpha^3 \\ \alpha^{10} & 0 & \alpha^4 & \alpha^4 & \alpha^1 \\ \alpha^9 & 0 & 1 & \alpha^1 & \alpha^4 \end{bmatrix}}_{\mathbf{W}}$$

This can be verified using Cook-Toom's decomposition with the same set of roots.

$$\underbrace{\begin{bmatrix} \alpha^5 & & & & \\ 1 & \alpha^5 & & & \\ & 1 & \alpha^5 & & \\ & & 1 & \alpha^5 & \\ & & & 1 & \end{bmatrix}}_{\mathbf{G}} = \underbrace{\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ \alpha^{13} & \alpha^{10} & \alpha^7 & \alpha^4 & \alpha^1 \\ \alpha^{11} & \alpha^5 & \alpha^{14} & \alpha^8 & \alpha^2 \\ \alpha^9 & 1 & \alpha^6 & \alpha^{12} & \alpha^3 \\ \alpha^7 & \alpha^{10} & \alpha^{13} & \alpha^1 & \alpha^4 \end{bmatrix}}_{\mathbf{C}} \underbrace{\begin{bmatrix} \alpha^1 & & & & \\ & 0 & & & \\ & & \alpha^4 & & \\ & & & \alpha^3 & \\ & & & & \alpha^{12} \end{bmatrix}}_{\mathbf{D}'} \underbrace{\begin{bmatrix} 1 & \alpha^2 & \alpha^4 & \alpha^6 \\ 1 & \alpha^5 & \alpha^{10} & 1 \\ 1 & \alpha^8 & \alpha^1 & \alpha^9 \\ 1 & \alpha^{11} & \alpha^7 & \alpha^3 \\ 1 & \alpha^{14} & \alpha^{13} & \alpha^{12} \end{bmatrix}}_{\mathbf{A}} \tag{4.6}$$

△

The next section shows how a filter bank can be constructed starting from Cook-Toom's algorithm.

4.3 Critically Subsampled Filter Bank implementing RS Codes

4.3.1 Algebraic Structure of RS Generator Matrix

First, the generator matrix $\mathbf{G}_{\mathcal{R}}$ can be diagonalized using a DFT decomposition (Cook-Toom's algorithm with interpolation points $\boldsymbol{\rho} = \left[\alpha^0 \alpha^1 \ldots \alpha^{\nu-1}\right]$):

$$\mathbf{G}_{\mathcal{R}} = \mathbf{F}_{\nu}^{-1} \boldsymbol{\Delta} \mathbf{F}_{\nu} \left[:, 0 : \kappa - 1\right] \tag{4.7}$$

Note that due to the choice of $\boldsymbol{\rho}$, we are dealing with a DFT-like decomposition which is made more explicit by the choice of $\mathbf{F}_{\nu}$ denoting a DFT matrix. By the definition of an RS code, $g_{\mathcal{R}}(z^{-1}) = 0$ at the interpolation points $\alpha^{b:b+\nu-\kappa-1}$. Hence, the diagonal matrix $\boldsymbol{\Delta}$ contains a block of $\nu - \kappa$ consecutive zeros at $\alpha^{b:b+\nu-\kappa-1}$.

For reasons that will become clear later, $\mathbf{\Delta}$ is cyclically shifted by $\tilde{b} = (\nu - \kappa + b) \bmod M$ steps yielding $\tilde{\mathbf{\Delta}}$:

$$\tilde{\mathbf{\Delta}} = \mathbf{R}_\nu^{-\tilde{b}} \mathbf{\Delta} \mathbf{R}_\nu^{\tilde{b}} \tag{4.8}$$

In this equation, $\mathbf{R}_x$ represents a rotation matrix:

$$\mathbf{R}_x = \begin{bmatrix} \mathbf{O}_{1\times x-1} & 1 \\ \mathbf{I}_{x-1} & \mathbf{O}_{x-1,1} \end{bmatrix}. \tag{4.9}$$

The generator matrix is now decomposed as follows:

$$\mathbf{G}_\mathcal{R} = \tilde{\mathbf{F}}_\nu^{-1} \tilde{\mathbf{\Delta}} \tilde{\mathbf{F}}_\nu \left[:, 0 : \kappa - 1\right] \tag{4.10}$$

with

$$\tilde{\mathbf{F}}_\nu = \mathbf{R}_\nu^{-\tilde{b}} \tilde{\mathbf{F}}_\nu \,. \tag{4.11}$$

This merely corresponds to a Cook-Toom decomposition with interpolation points $\tilde{\rho} = \alpha^{\tilde{b}:\tilde{b}+\nu-1}$.

In the second step, the diagonal matrix $\tilde{\mathbf{\Delta}}$ is decomposed as $\mathbf{S}\mathbf{G}_\mathcal{D}\mathbf{W}$ by applying the reverse Cook-Toom decomposition on a collection of submatrices. Therefore, define M diagonal submatrices

$$\underline{\tilde{\mathbf{\Delta}}}_m = \underline{\tilde{\mathbf{\Delta}}} \left[m : M : m + (\nu' - 1)M, m : M : m + (\nu' - 1)M\right], \forall m = 0 : M - 1 \tag{4.12}$$

and M subvectors

$$\underline{\tilde{\rho}}_m = \tilde{\rho} \left[m : M : m + (\nu' - 1)M\right] = \alpha^{\tilde{b}+m} \begin{bmatrix} \alpha^0 & \alpha^M & \dots & \alpha^{M(\nu'-1)} \end{bmatrix}, \tag{4.13}$$

holding the interpolation points used in the Cook-Toom decomposition. Note that the $\nu - \kappa$ zeros of the matrix $\tilde{\mathbf{\Delta}}$ are almost evenly distributed over the M submatrices, i.e. (since κ and M can be coprime) ζ submatrices $\underline{\tilde{\mathbf{\Delta}}}_m$ contain $\lfloor(\nu - \kappa)/M\rfloor$ zeros while $M - \zeta$ have $\lceil(\nu - \kappa)/M\rceil$ zeros, with $\zeta = M\lceil(\nu - \kappa)/M\rceil - \nu + \kappa$. Due to the rotation, it is seen that the first $m = 0 : \zeta - 1$ submatrices have one zero less than the subsequent ones. Otherwise stated, the number of zeros of the m-th submatrix $\underline{\tilde{\mathbf{\Delta}}}_m$ equals $L_m - 1$ with

$$L_m = \lfloor(m + 1)(\nu - \kappa)/M\rfloor - \lfloor m(\nu - \kappa)/M\rfloor + 1. \tag{4.14}$$

For each submatrix $\underline{\tilde{\mathbf{\Delta}}}_m$, the reverse Cook-Toom algorithm can now be applied with $\nu' = \frac{\nu}{M}$ interpolation points $\underline{\tilde{\rho}}_m$ and parameter L_m:

$$\underline{\mathbf{\Delta}}_m = \underline{\mathbf{S}}_m \underline{\mathbf{G}}_{m,\mathcal{D}} \underline{\mathbf{W}}_m, \forall m = 0 : M - 1. \tag{4.15}$$

With

$$\kappa'_m = \nu' - L_m + 1 \tag{4.16}$$

$\underline{\mathbf{G}}_{m,\mathcal{D}}$ is an $\nu' \times \kappa'_m$[1] Toeplitz matrix, holding a filter $d_m(z^{-1})$ of length L_m. Hence, using Theorem 4.1, the roots of $d_m(z^{-1})$ are

$$\alpha^{b+(m-\zeta \bmod M)+M(0:L_m-1)}, \forall m = 0 : M-1\,. \tag{4.17}$$

Now, one large matrix $\mathbf{G}_{\mathcal{D}}$ is constructed, containing $\underline{\mathbf{G}}_{m,\mathcal{D}}, \forall m = 0 : M-1$:

$$\mathbf{G}_{\mathcal{D}}\,[m : M : m + (\nu' - 1)M, m : M : k] = \underline{\mathbf{G}}_{m,\mathcal{D}}, \forall m = 0 : M-1\,. \tag{4.18}$$

In an analog way, the other submatrices $\underline{\mathbf{\Delta}}_m$, $\underline{\mathbf{S}}_m$ and $\underline{\mathbf{W}}_m$ are substituted yielding

$$\tilde{\mathbf{\Delta}} = \mathbf{S}\mathbf{G}_{\mathcal{D}}\mathbf{W}\,. \tag{4.19}$$

In the third step, $\mathbf{S}$ is studied in greater detail. Since M divides ν, subfields with M and ν' elements exist, defining the DFT matrices (in $\mathbb{F}_q$) $\mathbf{F}_M$ and $\mathbf{F}_{\nu'}$. Therefore, the $\nu \times \nu$ DFT matrix $\mathbf{F}_\nu$ can be decomposed with an FFT-like procedure into

$$\mathbf{F}_\nu = \mathbf{\Pi}_M(\mathbf{I}_M \otimes \mathbf{F}_{\nu'})\mathbf{\Pi}_{\nu'}\mathbf{Q}_{\nu'}(\mathbf{I}_{\nu'} \otimes \mathbf{F}_M)\mathbf{\Pi}_M\,. \tag{4.20}$$

Recall that $\mathbf{\Pi}_M$ and $\mathbf{\Pi}_{\nu'}$ are $\nu \times \nu$ interleaver matrices as defined in Eq. 3.19. Finally, the diagonal matrix

$$\mathbf{Q}_{\nu'} = \mathrm{Diag}\left(\alpha^{[0:\nu'-1]\otimes[0:M-1]}\right)\,. \tag{4.21}$$

contains the so-called "twiddle-factors". The rotated version of the DFT matrix $\tilde{\mathbf{F}}_\nu$ can also be decomposed in an FFT-like manner as follows:

$$\begin{aligned} \tilde{\mathbf{F}}_\nu &= \mathbf{R}_\nu^{-\tilde{b}}\mathbf{F}_\nu & (4.22)\\ &= \mathbf{\Pi}_M(\mathbf{I}_M \otimes \mathbf{F}_{\nu'})\mathbf{\Pi}_{\nu'}\tilde{\mathbf{Q}}_{\nu'}(\mathbf{I}_{\nu'} \otimes \tilde{\mathbf{F}}_M)\mathbf{\Pi}_M\,. & (4.23)\end{aligned}$$

In this equation, $\tilde{\mathbf{Q}}_{\nu'}$ is denoted as follows:

$$\tilde{\mathbf{Q}}_{\nu'} = \mathrm{Diag}\left(\alpha^{([0:\nu'-1])\otimes(\tilde{b}+[0:M-1])}\right)\,. \tag{4.24}$$

Returning to $\mathbf{S}$, note that $\underline{\mathbf{S}}_m = \mathbf{F}_{\nu'}\mathrm{Diag}\left(\alpha^{(m+\tilde{b})[0:\nu'-1]}\right)$. Essentially, $\mathbf{S}$ consists of M torn apart $\mathbf{F}_{\nu'}$ matrices, each multiplied by a diagonal matrix. Therefore, it is seen that

$$\mathbf{S} = \mathbf{\Pi}_M(\mathbf{I}_M \otimes \mathbf{F}_{\nu'})\mathbf{\Pi}_{\nu'}\mathbf{Q}_{\nu'}. \tag{4.25}$$

Stated otherwise,

$$\tilde{\mathbf{F}}_\nu = \mathbf{S}(\mathbf{I}_{\nu'} \otimes \tilde{\mathbf{F}}_M)\mathbf{\Pi}_M. \tag{4.26}$$

[1] κ'_m can be dependent on m if κ is not coprime with ν. Only if this last condition is met, $\kappa'_m = \frac{\kappa}{M}$ and is independent of m.

Substituting Eq. 4.19 in Eq. 4.10 and using Eq. 4.26 gives:

$$\mathbf{G}_{\mathcal{R}} = \tilde{\mathbf{F}}_{\nu}^{-1}\tilde{\mathbf{\Delta}}\tilde{\mathbf{F}}_{\nu}\,[0:\nu-1, 0:\kappa-1] \tag{4.27}$$

$$\mathbf{G}_{\mathcal{R}} = \tilde{\mathbf{F}}_{\nu}^{-1}\mathbf{S}\mathbf{G}_{\mathcal{D}}\underbrace{\mathbf{W}\tilde{\mathbf{F}}_{\nu}\,[0:\nu-1, 0:\kappa-1]}_{\mathbf{V}} \tag{4.28}$$

$$\mathbf{G}_{\mathcal{R}} = \mathbf{\Pi}_M^{-1}\underbrace{\left(\mathbf{I}_{\nu'}\otimes\tilde{\mathbf{F}}_M^{-1}\right)}_{\mathbf{G}_{\mathcal{C}}}\mathbf{G}_{\mathcal{D}}\underbrace{\mathbf{W}\tilde{\mathbf{F}}_{\nu}\,[0:\nu-1, 0:\kappa-1]}_{\mathbf{V}} \tag{4.29}$$

$$\boxed{\mathbf{G}_{\mathcal{R}} = \mathbf{\Pi}_{\nu'}\underbrace{\mathbf{G}_{\mathcal{C}}\mathbf{G}_{\mathcal{D}}}_{\mathbf{G}^{V}_{\mathcal{R}^{\Pi_{\nu'}}}}\mathbf{V}} \tag{4.30}$$

which is the desired decomposition.

To clarify the mathematics, an example is given first. Then, it is explained that the factors of $\mathbf{G}^{V}_{\mathcal{R}^{\Pi_{\nu'}}}$ indeed represent a filter bank like structure. Secondly, it is shown that $\mathbf{G}^{V}_{\mathcal{R}^{\Pi_{\nu'}}}$ implements an RS code.

Example 4.2 Let $\mathcal{R}\,[15, 10]$ denote the RS code introduced in Example 3.3 with $g_{\mathcal{R}}(z^{-1}) = \prod_{k=3}^{7}(z^{-1}-\alpha^k)$. In Chapter 3, it is explained how this code can be represented by a 5-band critically subsampled FIR filter bank. The method described in the previous chapter could be applied since the number of bands M divides κ and ν. The filter bank so obtained is shown in Figure 3.11. The techniques of this chapter will be illustrated on exactly the same code, but with $M = 3$ bands instead of 5, which excludes the application of the procedure of Chapter 3.

The decomposition of the Toeplitz generator matrix $\mathbf{G}_{\mathcal{R}}$ for $\mathcal{R}\,[15, 10]$ results in the following Δ:

$$\begin{aligned}\boldsymbol{\rho} \quad &\rightarrow \quad 1\ \ \alpha^1\,\alpha^2\,\alpha^3\,\alpha^4\,\alpha^5\,\alpha^6\,\alpha^7\,\alpha^8\ \ \alpha^9\ \ \alpha^{10}\,\alpha^{11}\,\alpha^{12}\,\alpha^{13}\,\alpha^{14}\\ \mathbf{\Delta} = &\mathrm{Diag}([\alpha^2\,\alpha^6\,\alpha^2\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \alpha^2\,\alpha^{11}\,\alpha^{12}\ \ 1\ \ \ \alpha^6\ \ \alpha^1\ 1]).\end{aligned}$$

In the next step, Δ is cyclically shifted by $\tilde{b} = 2$ steps:

$$\begin{aligned}\boldsymbol{\rho} \quad &\rightarrow \quad \alpha^2\,\alpha^3\,\alpha^4\,\alpha^5\,\alpha^6\,\alpha^7\,\alpha^8\ \ \alpha^9\ \ \alpha^{10}\,\alpha^{11}\,\alpha^{12}\,\alpha^{13}\,\alpha^{14}\ \ 1\ \ \ \alpha^1\\ \tilde{\mathbf{\Delta}} = &\mathrm{Diag}([\alpha^2\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \alpha^2\,\alpha^{11}\,\alpha^{12}\ \ 1\ \ \ \alpha^6\ \ \alpha^1\ \ \ 1\ \ \alpha^2\,\alpha^6]).\end{aligned}$$

In the second step, $\tilde{\mathbf{\Delta}}$ is decomposed as $\mathbf{S}\mathbf{G}_{\mathcal{D}}\mathbf{W}$. Therefore, define $M = 3$ submatrices $\underline{\tilde{\mathbf{\Delta}}}_m$, $m = 0 : M-1$:

$$\begin{aligned}\underline{\tilde{\mathbf{\Delta}}}_0 &= \mathrm{Diag}([\alpha^2\,0\ \alpha^2\ \ 1\ \ 1\ ])\ @\ \text{CT roots}\ [\alpha^2\,\alpha^5\ \ \alpha^8\ \ \alpha^{11}\,\alpha^{14}]\\ \underline{\tilde{\mathbf{\Delta}}}_1 &= \mathrm{Diag}([\ 0\ \ 0\,\alpha^{11}\,\alpha^6\,\alpha^2])\ @\ \text{CT roots}\ [\alpha^3\,\alpha^6\ \ \alpha^9\ \ \alpha^{12}\ \ 1\ \ ]\\ \underline{\tilde{\mathbf{\Delta}}}_2 &= \mathrm{Diag}([\ 0\ \ 0\,\alpha^{12}\,\alpha^1\,\alpha^6])\ @\ \text{CT roots}\ [\alpha^4\,\alpha^7\,\alpha^{10}\,\alpha^{13}\ \ \alpha^1\ ]\end{aligned}$$

Note that $\underline{\tilde{\mathbf{\Delta}}}_0$ has only one zero diagonal element. Hence, the reverse Cook-Toom algorithm (Theorem 4.1) can be applied with the corresponding roots, $M = 5$ and $L = 2$, resulting in the following decomposition

$$\underline{\tilde{\mathbf{\Delta}}}_0 = \underline{\mathbf{S}}_0 \underline{\mathbf{G}}_{0,\mathcal{D}} \underline{\mathbf{W}}_0. \tag{4.31}$$

Note that $\underline{\mathbf{G}}_{0,\mathcal{D}}$ is the 5×4 Toeplitz matrix, which was already given in Example 4.1. A similar decomposition holds for $\underline{\tilde{\mathbf{\Delta}}}_1$ and $\underline{\tilde{\mathbf{\Delta}}}_2$, where $\underline{\mathbf{G}}_{1,\mathcal{D}}$ and $\underline{\mathbf{G}}_{2,\mathcal{D}}$ are 5×3 Toeplitz matrices. Note that due to the rotation of $\mathbf{\Delta}$, the Toeplitz matrices $\underline{\mathbf{G}}_{m,\mathcal{D}}$ are ordered according to their size, starting with the largest. Substitution of the submatrices yields $\tilde{\mathbf{\Delta}} = \mathbf{S}\mathbf{G}_{\mathcal{D}}\mathbf{W}$. Looking at $\mathbf{G}_{\mathcal{D}}$ in more detail, it is seen that this matrix has a very specific structure (See also Figure 4.1):

$$\mathbf{G}_{\mathcal{D}}\,[0:3:12, 0:3:9] = \underline{\mathbf{G}}_{0,\mathcal{D}} \ (5 \times 4 \text{ Black boxes}) \tag{4.32}$$

$$\mathbf{G}_{\mathcal{D}}\,[1:3:13, 1:3:7] = \underline{\mathbf{G}}_{1,\mathcal{D}} \ (5 \times 3 \text{ White boxes}) \tag{4.33}$$

$$\mathbf{G}_{\mathcal{D}}\,[2:3:14, 2:3:8] = \underline{\mathbf{G}}_{2,\mathcal{D}} \ (5 \times 3 \text{ Gray boxes}) \tag{4.34}$$

Note that $\underline{\mathbf{S}}_m = \mathbf{F}_5 \mathrm{Diag}\left(\alpha^{(m+b)[0:4]}\right)$. Essentially, $\mathbf{S}$ consists of M torn apart $\mathbf{F}_5$ matrices, each multiplied by a diagonal matrix. Therefore, it is seen that

$$\mathbf{S} = \mathbf{\Pi}_3(\mathbf{I}_3 \otimes \mathbf{F}_5)\mathbf{\Pi}_5\tilde{\mathbf{Q}}_5. \tag{4.35}$$

△

4.3.2 Critically Subsampled Filter Bank based on Matrix Decompositions

Recall from Eq. 4.30 that

$$\mathbf{G}^{V}_{\mathcal{R}^{\Pi_{\nu'}}} = \mathbf{G}_{\mathcal{C}}\mathbf{G}_{\mathcal{D}}\,. \tag{4.36}$$

Referring to the discussion in the previous chapter, it is clear that $\mathbf{G}^{V}_{\mathcal{R}^{\Pi_{\nu'}}}$ is a generator matrix of an RS code, equivalent to $\mathcal{R}\,[\nu, \kappa]$. With a properly defined synthesis bank, which is slightly different compared to Theorem 3.2, the following theorem reveals a critically subsampled filter bank structure behind Eq. 4.30.

Theorem 4.2 *Assume a multirate system as depicted in Figure 3.1, critically subsampled by M. The analysis, respectively synthesis filters are defined as follows:*

$$a_m(z^{-1}) = z^{-(M-1-m)} \tag{4.37}$$

$$c_m(z^{-1}) = \sum_{k=0}^{M-1} \alpha^{-\nu'(m+\tilde{b})k} z^{-k} \tag{4.38}$$

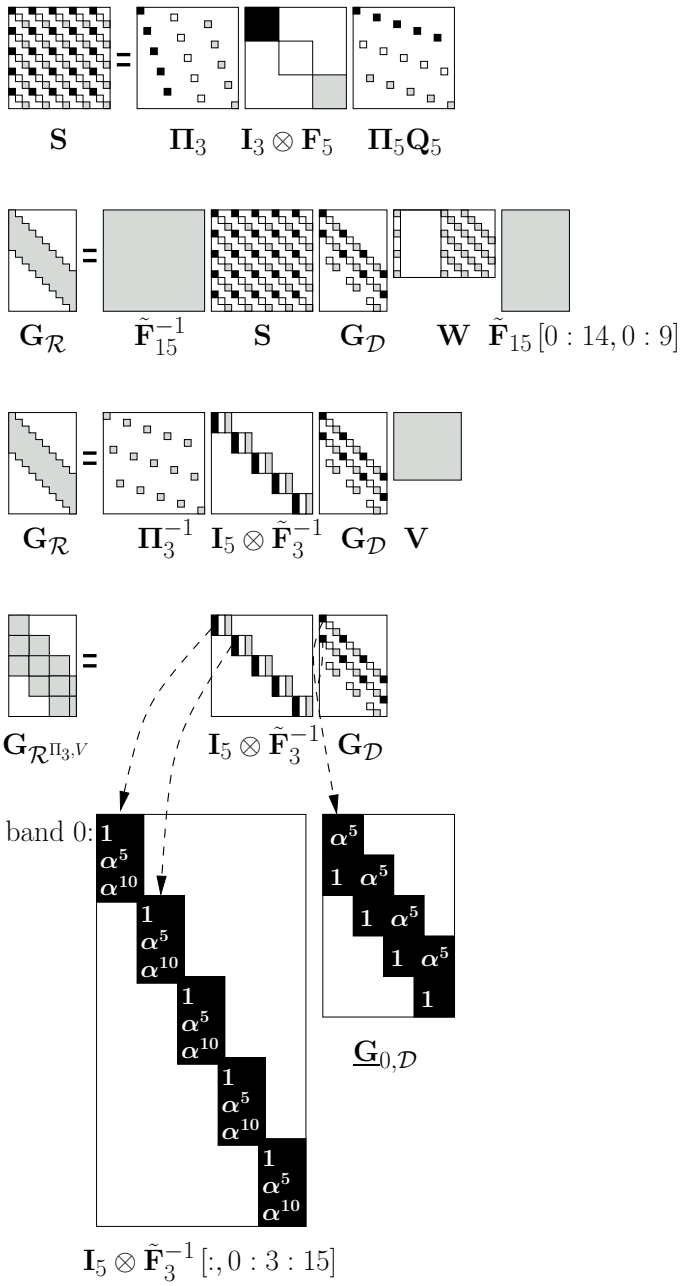


Figure 4.1: Factorization of the generator matrix of $\mathcal{R}\,[15, 10]$. The three middle factors define a critically subsampled filter bank implementing the $\mathcal{R}^{\Pi_3,V}\,[15, 10]$ code.

If the roots of the subband filters $d_m(z^{-1})$ are chosen as in equation 4.17, then this critically subsampled filter bank implements a code equivalent to $\mathcal{R}[\nu, \kappa]$.

This paragraph further clarifies why the factors of $\mathbf{G}^V_{\mathcal{R}^{\Pi_{\nu'}}}$ indeed represent a filter bank like structure as defined in the theorem: assume a block of κ data symbols $\mathbf{u}$ is fed into this filter bank. The analysis bank splits this block into M blocks. The m-th block contains the samples $\begin{bmatrix} u_m & u_{m+M} & u_{m+2M} & \dots \end{bmatrix}$. Again, note that since κ should not be a multiple of M, the first ζ blocks contain $\lceil \frac{\kappa}{M} \rceil$ data symbols while the remaining blocks have one sample less: $\lfloor \frac{\kappa}{M} \rfloor$ symbols.

Each of these blocks is consequently convolved with the corresponding subband filter $d_m(z^{-1})$. This is a terminated convolution, yielding *exactly* ν' subband samples in each band. Note that the upper ζ bands have a slightly shorter subband filter, compensating for a longer block of input samples. In a matrix form, these operations correspond to a multiplication with $\underline{\mathbf{G}}_{0,\mathcal{D}}$ for the first band (black boxes in Figure 4.1), a multiplication with $\underline{\mathbf{G}}_{1,\mathcal{D}}$ for the second band etc. The overall operation of the analysis bank and the subband filtering is seen to correspond with a multiplication with $\mathbf{G}_{\mathcal{D}}$.

The vector of subband samples will be referred to as $\mathbf{z}$, a $\nu \times 1$ vector. Hence,

$$\mathbf{z} = \mathbf{G}_{\mathcal{D}}\mathbf{u}\,. \tag{4.39}$$

Note that $\mathbf{G}_{\mathcal{D}}$ can be seen as the generator matrix of a code $\mathcal{D}$, which adds all the redundancy (from dimension κ to ν). However, $\mathbf{G}_{\mathcal{D}}$ is not yet a Reed-Solomon code; as in the previous chapter, the synthesis bank is crucial to combine the subband samples into a valid codeword – as will be shown. More specifically, the subband samples of band m are filtered with the m-th column of the *rotated* DFT-matrix $\tilde{\mathbf{F}}_M^{-1}$ (black boxes in Figure 4.1). However, the subband samples are upsampled with a factor M first resulting in a stretched Toeplitz matrix. Together with the other bands, the operations performed by the synthesis bank correspond to a multiplication with $\mathbf{I}_{\nu'} \otimes \tilde{\mathbf{F}}_M^{-1}$. Hence, the output of the filter bank $\mathbf{y}$ (codeword of $\mathcal{R}^{\Pi_{\nu'}}[\nu, \kappa]$) can be written as:

$$\mathbf{y} = \mathbf{G}_{\mathcal{C}}\mathbf{z}\,. \tag{4.40}$$

Example 4.3 Figure 4.1 shows the factors of $\mathbf{G}^V_{\mathcal{R}^{\Pi_{\nu'}}}$ for the RS code $\mathcal{R}[15, 10]$. The filter bank representation itself is shown in Figure 4.2. Considering only the first subband, the correspondence between the filtering with $d_0(z^{-1})$ and the multiplication with the corresponding (indicated in black) submatrix of $\mathbf{G}_{\mathcal{D}}$ is indicated. This filter delivers 5 samples at the output, while the input consists of only 4 samples (see Eq. 4.31). Finally, in the synthesis bank, the samples are upsampled and filtered with $1 + \alpha^5 z^{-1} + \alpha^{10} z^{-2}$ (represented by $(\mathbf{I}_5 \otimes \tilde{\mathbf{F}}_3^{-1})[:, 0:3:15]$). Adding the corresponding outputs of the other bands

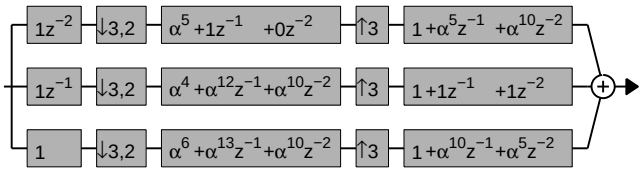


Figure 4.2: Critically subsampled filter bank of $\mathcal{R}\,[15, 10]$ as discussed in Example 4.3

gives the codeword. Due to the critical subsampling, there are as many subband samples as codeword samples. △

4.3.3 Circulant Block Structure of $\mathcal{R}^{\Pi_{\nu'}}\,[\nu, \kappa]$

In the previous chapter, a $\nu \times \kappa$ block Toeplitz matrix with $M \times M$ circulant blocks plays a central role in the decomposition. This section shows that an analog structure exists if κ is not a multiple of M. Therefore, a more complex analysis bank is called upon, which can be obtained by decomposing the $\kappa \times \kappa$ basis transformation matrix $\mathbf{V}$ as follows:

$$\mathbf{V} = \mathbf{G}_{\mathcal{A}}\mathbf{V}' \tag{4.41}$$

with

$$\mathbf{G}_{\mathcal{A}} = \mathbf{I}_{\nu'} \otimes \tilde{\mathbf{F}}_M\,[0 : \kappa - 1, 0 : \kappa - 1]\ .$$

It is seen that this decomposition is a filter bank-like decomposition as depicted in Figure 3.1, but with the analysis bank filters changed as follows:

$$a_m(z^{-1}) = \sum_{k=0}^{M-1} \alpha^{\nu'(m+\tilde{b})(M-1-k)} z^{-k}. \tag{4.42}$$

The analysis bank performs some sort of precoding on the dataword such that the code generated by this filter bank is $\mathcal{R}\,[\nu, \kappa]$.

Using the fact that the eigenvectors of a circulant matrix are the columns of a DFT matrix, this generator matrix is found to be a block Toeplitz matrix with

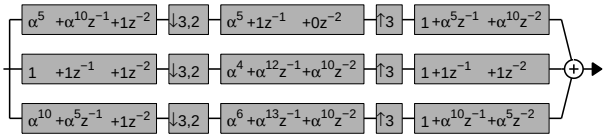


Figure 4.3: Critically subsampled filter bank ($M = 3$) for the $\mathcal{R}[15, 10]$ code described in Example 4.4.

circulant blocks (except for the last block column).

$$\mathbf{G}_{\mathcal{R}^{\Pi_{M,V'}}} = \begin{bmatrix} \mathbf{C}_0 & & & & \\ \mathbf{C}_1 & \mathbf{C}_0 & & & \\ \mathbf{C}_2 & \mathbf{C}_1 & \ddots & & \\ \vdots & \mathbf{C}_2 & \ddots & \mathbf{C}_0 & \\ \mathbf{C}_K & \vdots & \ddots & \mathbf{C}_1 & \mathbf{C'}_0 \\ & \mathbf{C}_K & & \mathbf{C}_2 & \mathbf{C'}_1 \\ & & \ddots & \vdots & \vdots \\ & & & \mathbf{C}_K & \mathbf{C'}_{K-1} \end{bmatrix} \tag{4.43}$$

with

$$\begin{aligned} \mathbf{C}_k &= \tilde{\mathbf{F}}_M^{-1} \mathbf{D}_k \tilde{\mathbf{F}}_M, \quad \forall k = 0 : \lceil 1 + (\nu - \kappa)/M \rceil - 1 \\ \mathbf{D}_k &= \underline{\mathbf{G}}_{\mathcal{D}} [kM : kM + M - 1, 0 : M - 1] \, . \end{aligned}$$

The last $M' = k \bmod M$ columns are somewhat special. They are

$$\begin{aligned} \mathbf{C'}_k &= \tilde{\mathbf{F}}_M^{-1} \mathbf{D'}_k \tilde{\mathbf{F}}_M [:, 0 : M' - 1] \text{ with} \\ \mathbf{D'}_k &= \begin{bmatrix} \mathbf{D}_k [0 : M' - 1, 0 : M' - 1] & \mathbf{0}_{M' \times M - M'} \\ \mathbf{0}_{M - M' \times M'} & \mathbf{0}_{M - M' \times M - M'} \end{bmatrix} . \end{aligned}$$

Hence, $\mathbf{C'}_k$ can be viewed as a sort of *best rank M' approximation* of $\mathbf{C}_k$. This is related to earlier work published by Solomon and Van Tilborg [68]. In this paper, the generator matrix of an RS code is transformed into a block matrix with circulant blocks. However, their approach does not apply when κ and M are coprime. Here, it is shown that even with κ and M coprime, this quasi-cyclic representation exists.

Example 4.4 For the $\mathcal{R}[15, 10]$ defined in Example 4.2, the filter bank representation with the analysis bank defined according to Eq. 4.41 is shown in

Figure 4.3. The generator matrix is structured as follows:

$$\mathbf{G}_{\mathcal{R}^{\Pi_{\nu'}}}^{V'} = \begin{bmatrix} \mathbf{C}_0 & & & \\ \mathbf{C}_1 & \mathbf{C}_0 & & \\ \mathbf{C}_2 & \mathbf{C}_1 & \mathbf{C}_0 & \\ & \mathbf{C}_2 & \mathbf{C}_1 & \mathbf{C'}_0 \\ & & \mathbf{C}_2 & \mathbf{C'}_1 \end{bmatrix} \tag{4.44}$$

with

$$\mathbf{C}_0 = \begin{bmatrix} \alpha^{14} & \alpha^6 & \alpha^5 \\ \alpha^5 & \alpha^{14} & \alpha^6 \\ \alpha^6 & \alpha^5 & \alpha^{14} \end{bmatrix} \mathbf{C}_1 = \begin{bmatrix} \alpha^4 & 0 & \alpha^6 \\ \alpha^6 & \alpha^4 & 0 \\ 0 & \alpha^6 & \alpha^4 \end{bmatrix} \tag{4.45}$$

$$\mathbf{C}_2 = \begin{bmatrix} 0 & \alpha^5 & \alpha^0 \\ \alpha^0 & 0 & \alpha^5 \\ \alpha^5 & \alpha^0 & 0 \end{bmatrix} \mathbf{C'}_0 = \begin{bmatrix} \alpha^5 \\ \alpha^{10} \\ \alpha^0 \end{bmatrix} \mathbf{C'}_1 = \begin{bmatrix} \alpha^0 \\ \alpha^5 \\ \alpha^{10} \end{bmatrix}. \tag{4.46}$$

Finally, the structure of the matrices is depicted in Figure 4.4. △

Note that the construction of an equivalent RS code with a block generator matrix with circulant blocks relies on the exploitation of the cyclic properties of the RS code. The fundamental operation that transforms the cyclic character modulo the block length (ν) into a cyclic behavior modulo a divisor of this block length (M) is the FFT (Eq. 4.20).

4.4 Filter Bank Representations for BCH Codes

In the previous section, it is shown how the cyclic character of the RS code modulo $1 + x^\nu$ is transformed by the filter bank into a cyclic character modulo $1 + x^M$, with $M|\nu$. This condition is seen to be a crucial element in the derivation since, for RS codes, $\nu = q - 1$ and thus $M|q - 1$. The latter allows an M-point DFT to exist in $\mathbb{F}_q$. For the more general family of BCH codes, ν can differ from $q-1$ such that $M|\nu$ no longer guarantees the existence of an M-point DFT transform in $\mathbb{F}_q$. This section deals with filter bank representations for BCH codes.

4.4.1 Construction of the Filter Bank in an Extension Field

Let $\mathcal{B}[\nu, \kappa]$ be a BCH code in $\mathbb{F}_q$. Let n be the *multiplicative order of q modulo* ν, i.e. n is the smallest integer such that $x^\nu - 1 | x^{q^n-1} - 1$. Let $\alpha \in \mathbb{F}_{q^n}$ be a primitive ν-th root of unity in $\mathbb{F}_{q^n}$. Let M be a common divisor of ν and $q^n - 1$ with M and $q - 1$ coprime. Unless ν is prime, this is always possible

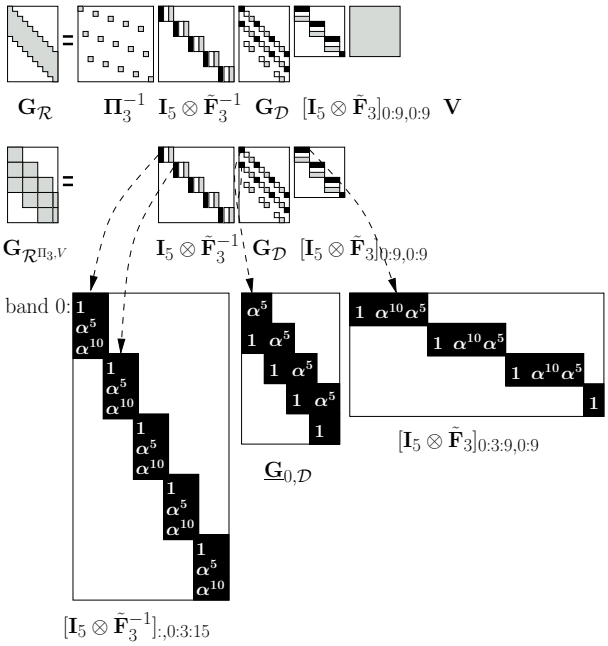


Figure 4.4: Factorization of the generator matrix of $\mathcal{R}\,[15, 10]$. The three middle factors define a critically subsampled filter bank implementing the $\mathcal{R}^{\Pi_3}\,[15, 10]$ RS code.

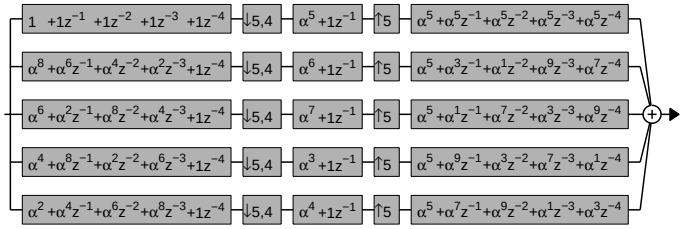


Figure 4.5: Critically subsampled filter bank representation over $\mathbb{F}_{q^n}$ of the BCH code $\mathcal{B}[10, 5, 6]$

since $\nu | q^n - 1$. In the extension field $\mathbb{F}_{q^n}$, the M-point DFT transform exists and the filter bank can readily be constructed in the extension field, according to Theorem 4.2. Let the analysis bank be defined by Eq. 4.42, representing a DFT bank instead of a demultiplexer.

Example 4.5 Let us proceed with the BCH code $\mathcal{B}[10, 5, 6]$ of Example 2.8. The techniques presented previously can directly be applied to BCH codes. The filter coefficients in the resulting filter bank are powers of α. Unfortunately, $\alpha \notin \mathbb{F}_q$, as was the case for RS codes, where the primitive element $\beta \in \mathbb{F}_q$ equals the primitive ν-th root of unity α. So, unlike RS codes, by applying the techniques presented previously, we readily arrive at the filter bank representation of $\mathcal{B}[\nu, \kappa]$ in $\mathbb{F}_{q^n}$ (Figure 4.5). △

In the upcoming sections, it is shown that a filter bank in $\mathbb{F}_q$ can be obtained starting from the filter bank in the extension field $\mathbb{F}_{q^n}$.

4.4.2 Transforming the Filter Bank from $\mathbb{F}_{q^n}$ to $\mathbb{F}_q$

Before tackling this problem for a filter bank, let us first have a look how a single element of $\mathbb{F}_{q^n}$ can be decomposed into elements of $\mathbb{F}_q$. As suggested in Section 2.1, $\mathbb{F}_{p^m}$ can be seen as a vector space of dimension m over $\mathbb{F}_p$. Any set λ of linearly independent elements of $\mathbb{F}_p$ can serve as a basis for $\mathbb{F}_{p^m}$. If the field is constructed using its primitive polynomial $\mathscr{P}(x)$, the normal basis $\boldsymbol{\lambda} = \begin{bmatrix} 1 & \beta & \beta^2 & \dots \end{bmatrix}$ is used. However, other bases can be used. What is needed is a mathematical tool that allows us to easily decompose elements of a Galois field along a specified basis. This tool is called the *trace*.

Definition 4.3 (Trace) The trace of $a \in \mathbb{F}_{q^n}$ from $\mathbb{F}_{q^n}$ to $\mathbb{F}_q = \mathbb{F}_{p^m}$ is defined as

$$\mathrm{Tr}_n(a) = \sum_{j=0}^{n-1} a^{q^j} . \tag{4.47}$$

□

The trace of an element a always belongs to $\mathbb{F}_q$. This is a useful property for decomposing an element a along a specified base $\boldsymbol{\lambda} = \begin{bmatrix}\lambda(0) & \dots & \lambda(n-1)\end{bmatrix}$, as we will see. First, we define the complementary basis. A basis $\overline{\boldsymbol{\lambda}} = \begin{bmatrix}\overline{\lambda}[0] & \dots & \overline{\lambda}[n-1]\end{bmatrix}$ is said to be complementary to $\boldsymbol{\lambda}$ if

$$\mathrm{Tr}_n\left(\overline{\lambda}[i]\,\lambda[i]\right) = \delta_{ij}\,. \tag{4.48}$$

with δ_{ij} the Kronecker delta. Each element $a \in \mathbb{F}_{q^n}$ can now be written as [67]

$$a = \sum_{i=0}^{n-1} a^{\{i\}}\lambda[i] \tag{4.49}$$

with

$$a^{\{i\}} = \mathrm{Tr}_n\left(a\overline{\lambda}[i]\right) \in \mathbb{F}_q\,. \tag{4.50}$$

Two properties of the trace are called upon later on in this work:

$$\begin{aligned}\mathrm{Tr}_n(a+b) &= \mathrm{Tr}_n(a) + \mathrm{Tr}_n(b)\,, \forall a,b \in \mathbb{F}_{q^n} \\ \mathrm{Tr}_n(a\cdot b) &= a\cdot\mathrm{Tr}_n(b)\,, \forall a \in \mathbb{F}_q, b \in \mathbb{F}_{q^n}\,.\end{aligned} \tag{4.51}$$

Example 4.6 As an example, the trace can be used to obtain the polynomial representation of e.g. $a = \beta^5 \in \mathbb{F}_{3^2}$ defined by $\mathscr{P}(x) = x^2 + 2x + 2$. The complementary basis $\overline{\boldsymbol{\lambda}}$ of the normal base $\boldsymbol{\lambda} = \begin{bmatrix}1 & \beta\end{bmatrix}$ is calculated:

$$\overline{\boldsymbol{\lambda}} = \begin{bmatrix}\beta & \beta^2\end{bmatrix}\,.$$

Now, the traces are calculated according to Eq. 4.47:

$$\begin{aligned}\mathrm{Tr}_2\left(\overline{\lambda}_0 a\right) &= \mathrm{Tr}_2\left(\beta^1\cdot\beta^5\right) = \beta^1\cdot\beta^5 + \beta^{1^3}\cdot\beta^{5^3} = 0 \\ \mathrm{Tr}_2\left(\overline{\lambda}_1 a\right) &= \mathrm{Tr}_2\left(\beta^2\cdot\beta^5\right) = \beta^2\cdot\beta^5 + \beta^{2^3}\cdot\beta^{5^3} = \beta^4 = 2\end{aligned}$$

According to Eq. 4.49, $\beta^5 = 0\cdot\beta^0 + 2\cdot\beta^1$ which can be verified in Table 2.2. △

Next, it is explained how elements in $\mathbb{F}_{q^n}$ can be multiplied using $\mathbb{F}_q$ arithmetic. This leads to an extension of the concept of a trace. Assume $a, b, c = a\cdot b \in \mathbb{F}_{q^n}$. Using Eq. 4.49, a and b can be decomposed as follows:

$$\begin{aligned}a &= \sum_{i=0}^{n-1} a^{\{i\}}\lambda[i] \\ b &= \sum_{i=0}^{n-1} b^{\{i\}}\lambda[i]\,.\end{aligned}$$

Using Eq. 4.50 and assuming a normal basis ($\overline{\lambda}[i] = \gamma^i$) the i-th coordinate of c

$$\begin{aligned} c^{\{i\}} &= \mathrm{Tr}_n\left(a \cdot b\overline{\lambda}[i]\right) && (4.52)\\ &= \mathrm{Tr}_n\left(\overline{\lambda}[i]\sum_{j=0}^{n-1}\sum_{k=0}^{n-1} b^{\{j\}}a^{\{k\}}\gamma^{k+j}\right) && (4.53)\\ &= \sum_{j=0}^{n-1} b^{\{j\}}\sum_{k=0}^{n-1} a^{\{k\}}\mathrm{Tr}_n\left(\overline{\lambda}[i]\,\gamma^{k+j}\right). && (4.54) \end{aligned}$$

The last equation is obtained using the properties of the trace in Equations 4.51. The inner sum will be denoted in a special way

$$\begin{aligned} a^{\{i,j\}} &= \sum_{k=0}^{n-1} a^{\{k\}}\mathrm{Tr}_n\left(\overline{\lambda}[i]\,\gamma^{k+j}\right)\\ &= \sum_{k=0}^{n-1} \mathrm{Tr}_n\left(a^{\{k\}}\overline{\lambda}[i]\,\gamma^{k+j}\right)\\ &= \mathrm{Tr}_n\left(\sum_{k=0}^{n-1} a^{\{k\}}\overline{\lambda}[i]\,\gamma^{k+j}\right)\\ &= \mathrm{Tr}_n\left(\overline{\lambda}[i]\,\gamma^{j}\sum_{k=0}^{n-1} a^{\{k\}}\gamma^{k}\right)\\ &= \mathrm{Tr}_n\left(\overline{\lambda}[i]\,\gamma^{j}a\right). \end{aligned}$$

Using this notation, Eq. 4.52 becomes

$$c^{\{i\}} = \sum_{j=0}^{n-1} a^{\{i,j\}}b^{\{j\}},$$

which resembles a matrix multiplication. Indeed, defining the $n \times 1$ vectors $\mathbf{c}$ and $\mathbf{b}$ and the $n \times n$ matrix $\mathbf{A}$ as $c[i] = c^{\{i\}}$, $b[i] = b^{\{i\}}$ and $A[i,j] = a^{\{i,j\}}$,

$$\mathbf{c} = \mathbf{A}\mathbf{b}.$$

Example 4.7 Let us define $\mathbb{F}_{3^4}$ by its primitive Conway polynomial $\mathscr{P}(x) = 2 + 2x^3 + x^4$, with root γ such that $\beta = \gamma^{10} \in \mathbb{F}_{3^2}$ (see also Example 2.6). With $a = \gamma^{11}$, $b = \gamma^{23}$, c becomes γ^{34}. The complementary basis $\overline{\boldsymbol{\lambda}}$ of the normal base $\boldsymbol{\lambda} = \begin{bmatrix}1 & \gamma\end{bmatrix}$ is calculated:

$$\overline{\boldsymbol{\lambda}} = \begin{bmatrix}\gamma^{14} & \gamma^{45}\end{bmatrix}.$$

The elements a and b can be expanded according to this basis resulting in the following vectors/matrices:

$$\underbrace{\begin{bmatrix} \beta^7 \\ \beta^2 \end{bmatrix}}_{\mathbf{c}} = \underbrace{\begin{bmatrix} 0 & \beta^6 \\ \beta^1 & \beta^4 \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} \beta^2 \\ \beta^1 \end{bmatrix}}_{\mathbf{b}}$$

As can be verified, $\boldsymbol{\lambda}\mathbf{c} = c = \gamma^{34}$. △

All the notation is now defined to properly state the theorem:

Theorem 4.4 *Let $\mathcal{B}[\nu, \kappa]$ be a BCH code. Let $\alpha \in \mathbb{F}_{q^n}$ be a primitive ν-th root of unity. M is a common divisor of ν and $q^n - 1$. Let $a_m(z^{-1})$, $d_m(z^{-1})$ and $c_m(z^{-1})$ be the analysis, subband and synthesis filters of a critically subsampled filter bank over $\mathbb{F}_{q^n}$, as defined by the equations 4.42, 4.17 and 4.38, respectively. The $\mathcal{B}[\nu, \kappa]$ can be implemented as a* sum of n critically subsampled filter banks over $\mathbb{F}_q$. *The analysis, resp. synthesis bank of the n'-th filter bank ($n' = 0 : n-1$, band m) defined as ${a_m}^{\{n',0\}}(z^{-1})$, resp. ${c_m}^{\{0,n'\}}(z^{-1})$. The subband filters $\tilde{d}_m(z^{-1})$ are the same for each filter bank:*[2]

$$\tilde{d}_m(z^{-1}) = \sum_{k=0}^{n-1} a[k]\, d_m^{q^k}(z^{-1}) \tag{4.55}$$

with $a[k]$ a solution of the following system:

$$\begin{bmatrix} \overline{\lambda}[0] & \overline{\lambda}[0]^q & \overline{\lambda}[0]^{q^2} & \cdots & \overline{\lambda}[0]^{q^{n-1}} \\ \overline{\lambda}[0]^{q^{n-1}} & \overline{\lambda}[0] & \overline{\lambda}[0]^q & \cdots & \overline{\lambda}[0]^{q^{n-2}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \overline{\lambda}[0]^q & \overline{\lambda}[0]^{q^2} & \overline{\lambda}[0]^{q^3} & \cdots & \overline{\lambda}[0] \end{bmatrix} \begin{bmatrix} a[0] \\ a[1] \\ \vdots \\ a[n-1] \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \tag{4.56}$$

Proof: Considering an input $u(z^{-1}) = z^{j'}$ with $j' = 0 : N-1$, the filter bank output (impulse response) for $\mathcal{B}[\nu, \kappa]$ can be written as

$$y(z^{-1}) = \sum_{m=0}^{M-1} c_m(z^{-1}) d_m(z^{-N}) x_m(z^{-1})$$

with

$$x_m(z^{-1}) = \alpha^{\nu' m(N-1-j')}$$

[2] Given a polynomial $a(z^{-1})$, $a^b(z^{-1})$ denotes the polynomial with each coefficient raised to the power b.

We show that the filter bank output $y'(z^{-1})$ of the filter bank in $\mathbb{F}_q$ equals $y(z^{-1})$. The filter bank output of the n'-th filter bank of $\mathcal{B}[\nu, \kappa]$ equals

$$y_{n'}(z^{-1}) = \sum_{m=0}^{M-1} c_m^{\{0,n'\}}(z^{-1}) \tilde{d}_m(z^{-N}) x_m^{\{n',0\}}(z^{-1}) .$$

Substitutions of $\tilde{d}_m(z^{-1})$ (equation 4.55) and

$$\begin{aligned} c_m^{\{0,n'\}}(z^{-1}) &= \sum_{i=0}^{n-1} (c_m(z^{-1}) \overline{\lambda}[0] \rho^{n'})^{q^i} \\ x_m^{\{n',0\}}(z^{-1}) &= \sum_{j=0}^{n-1} (\overline{\lambda}[n'] x_m)^{q^j} \end{aligned}$$

and summing over all filter banks ($n' = 0 \ldots n-1$) while using

$$\sum_{n'=0}^{n-1} \gamma^{n' q^i} \overline{\lambda}[n']^{q^j} = \delta_{ij}$$

(Eq. 4.48) leads to

$$y(z^{-1}) = \sum_{m=0}^{M-1} \sum_{k=0}^{n-1} a[k] \sum_{i=0}^{n-1} \overline{\lambda}[0]^{q^i} c_m^{q^i}(z^{-1}) d_m^{q^k}(z^{-N}) x_m^{q^i}(z^{-1}) .$$

Grouping terms with $k - j$ constant ($i = k - j \bmod n$) gives:

$$y(z^{-1}) = \sum_{k=0}^{n-1} \sum_{j=0}^{n-1} a[k] \overline{\lambda}[0]^{q^{k-j}} \sum_{m=0}^{M-1} c_m^{q^{k-j}}(z^{-1}) d_m^{q^k}(z^{-N}) x_m^{q^{k-j}}(z^{-1}) .$$

Looking more in detail at the inner sum with $j = 0$, it can be seen that it is independent of k:

$$\sum_{m=0}^{M-1} c_m^{q^k}(z^{-1}) d_m^{q^k}(z^{-N}) x_m^{q^k}(z^{-1}) = y(z^{-1}) .$$

For $j \neq 0$, it can be seen that the inner sum is again independent of j. Summing over all i, $y(z^{-1}) = y'(z^{-1})$ if $a[k]$ is a solution of the system in equation 4.56 which proves the theorem. □

Example 4.8 In this example, the filter bank over $\mathbb{F}_{3^4}$ as shown in Figure 4.5 is transformed into a filter bank over $\mathbb{F}_{3^2}$ as stated by the previous theorem. In this case, the system of equations in 4.56 becomes

$$\begin{bmatrix} \alpha^{14} & \alpha^{46} \\ \alpha^{46} & \alpha^{14} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} , \tag{4.57}$$

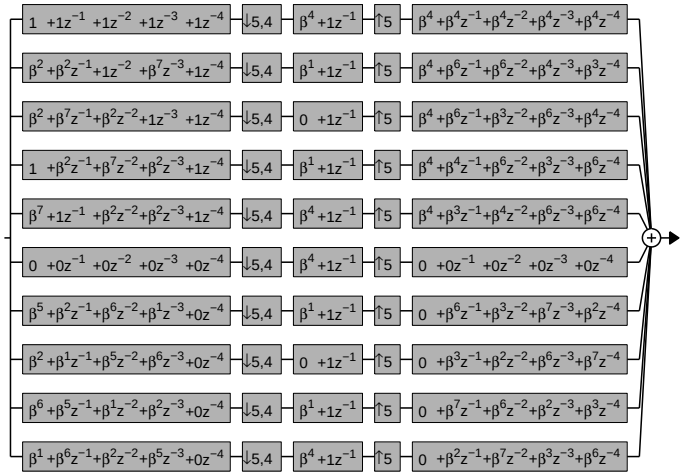


Figure 4.6: Critically subsampled filter bank representation over $\mathbb{F}_q$ of the BCH code $\mathcal{B}[10, 5, 6]$ ($n = 2$)

with solution $\mathbf{a} = \begin{bmatrix} \alpha^{59} \\ \alpha^{51} \end{bmatrix}$. The filter bank so obtained can be found in Figure 4.6. The operation underlying the existence of the filter bank is a decomposition of an $M \times M$ circulant matrix over $\mathbb{F}_q$ into a block diagonal $n \times n$ matrix with $M \times M$ diagonal blocks (over $\mathbb{F}_q$). This decomposition is 'special' in the sense that all the off-diagonal block matrices are zero, which is necessary for this decomposition to be used as a filter bank presentation.

$$\begin{bmatrix} \beta^7 & \beta^5 & 1 & \beta^5 & \beta^7 \\ \beta^7 & \beta^7 & \beta^5 & 1 & \beta^5 \\ \beta^5 & \beta^7 & \beta^7 & \beta^5 & 1 \\ 1 & \beta^5 & \beta^7 & \beta^7 & \beta^5 \\ \beta^5 & 1 & \beta^5 & \beta^7 & \beta^7 \end{bmatrix} = \left[\begin{array}{ccccc|ccccc} \beta^4 & \beta^4 & \beta^4 & \beta^4 & \beta^4 & 0 & 0 & 0 & 0 & 0 \\ \beta^4 & \beta^6 & \beta^6 & \beta^4 & \beta^3 & 0 & \beta^6 & \beta^3 & \beta^7 & \beta^2 \\ \beta^4 & \beta^6 & \beta^3 & \beta^6 & \beta^4 & 0 & \beta^3 & \beta^2 & \beta^6 & \beta^7 \\ \beta^4 & \beta^4 & \beta^6 & \beta^3 & \beta^6 & 0 & \beta^7 & \beta^6 & \beta^2 & \beta^3 \\ \beta^4 & \beta^3 & \beta^4 & \beta^6 & \beta^6 & 0 & \beta^2 & \beta^7 & \beta^3 & \beta^6 \end{array}\right] \cdot \ldots$$

$$\left[\begin{array}{ccccc|ccccc} \beta^4 & & & & & & & & & \\ & \beta^1 & & & & & & & & \\ & & 0 & & & & & & & \\ & & & \beta^1 & & & & & & \\ & & & & \beta^4 & & & & & \\ \hline & & & & & \beta^4 & & & & \\ & & & & & & \beta^1 & & & \\ & & & & & & & 0 & & \\ & & & & & & & & \beta^1 & \\ & & & & & & & & & \beta^4 \end{array}\right] \cdot \left[\begin{array}{ccccc} 1 & 1 & 1 & 1 & 1 \\ 1 & \beta^7 & 1 & \beta^2 & \beta^2 \\ 1 & 1 & \beta^2 & \beta^7 & \beta^2 \\ 1 & \beta^2 & \beta^7 & \beta^2 & 1 \\ 1 & \beta^2 & \beta^2 & 1 & \beta^7 \\ \hline 0 & 0 & 0 & 0 & 0 \\ 0 & \beta^1 & \beta^6 & \beta^2 & \beta^5 \\ 0 & \beta^6 & \beta^5 & \beta^1 & \beta^2 \\ 0 & \beta^2 & \beta^1 & \beta^5 & \beta^6 \\ 0 & \beta^5 & \beta^2 & \beta^6 & \beta^1 \end{array}\right]$$

△

4.5 Conclusion

This chapter shows that an M band critically subsampled filter bank representation exists for each RS code as long as $M|\nu$. Previous constraints on the dimension κ of the RS code are removed, such that there is flexibility in the choice of the code rate. The discussion here is mainly presented from a matrix decomposition point of view, since this representation is more suited for building filter banks for an entire gamut of RS codes. The derivation is based on two matrix decompositions: the reverse Cook-Toom algorithm and the Fast Fourier Transform (FFT). Concerning the latter, note that an M-point DFT transform is guaranteed to exist since $M|\nu$ implies that $M|q-1$. This is not true in the case of BCH codes in which ν can differ from $q-1$. Nevertheless, it is shown that a BCH code can be represented as a sum of n critically subsampled filter banks. Starting from a filter bank representation in an extension field $\mathbb{F}_{q^n}$, a transformation from $\mathbb{F}_{q^n}$ to $\mathbb{F}_q$ is applied yielding the desired result. This transformation is based on (an extension of) the trace operation.

Chapter 5

2-D Cyclic Codes and Filter Banks

5.1 Introduction

The main goal of this chapter is to look at the filter bank decompositions of RS codes from a coding theoretic perspective, opposed to the signal processing point of view taken in the previous chapters. More specifically, in this chapter, the filter bank structure of RS and BCH codes is seen from the point of view of a 2-D cyclic code. The material presented in this chapter is not crucial for understanding the rest of the thesis, and can be skipped at a first reading.

In a 2-D code, each codeword can be written as a matrix or a bivariate (2-dimensional) polynomial. 2-D codes can be obtained as a concatenated code, in which an outer code in a (large) field $\mathbb{F}_{q^\kappa}$ is concatenated with an inner code. The inner code (in $\mathbb{F}_q$) has dimension κ (q^κ codewords), such that the inner code maps each codeword symbol of the outer code onto κ symbols in $\mathbb{F}_q$. Such a concatenated code is commonly referred to as a concatenated code in the sense of Forney [27]. If the inner codewords are visualized as (κ dimensional) column vectors, they can be stacked into a matrix which is the codeword matrix of the corresponding 2-D code.

More interestingly, if inner and outer code are cyclic, we will show in this chapter that a 2-D *cyclic* code can be obtained. I.e. a cyclic shift of rows and/or columns of a codeword in matrix format gives another codeword.

This chapter shows there is a tight connection between the filter bank representations of RS codes and 2-D cyclic codes. A slight modification of the filter

bank decomposition (called type II) gives rise to a direct sum of concatenated (inner and outer) *cyclic* codes as described above. The outer codes correspond with the subband filters, while the inner codes correspond with the synthesis filters. It is also shown that each nonzero of the RS code is realized by a well defined combination of a nonzero of inner and outer code.

This chapter starts by explaining some basic concepts needed in Section 5.2. In Section 5.3, the correspondence between filter banks of RS codes and 2-D cyclic codes is discussed. These additional insights lead to a number of interesting applications in Section 5.4. In Section 5.4.1, it is seen how the filter bank theory can be applied recursively on the subband codes (which are seen to be cyclic). Secondly, Section 5.4.2 illustrates how the theory of 2-D cyclic codes extends towards BCH codes. Finally, in Section 5.4.3, we have a look at unequal error protecting codes.

5.2 Minimal Ideal and Idempotent

Before looking into the theory on 2-D cyclic codes, some more advanced properties related to cyclic codes are discussed. A cyclic code $\mathcal{I}[\nu,\kappa]$ can also be defined using its *idempotent* $i_{\mathcal{I}}(x)$, which is the unique codeword $i_{\mathcal{I}}(x) \in \mathcal{I}[\nu,\kappa]$ that generates $\mathcal{I}[\nu,\kappa] = < i_{\mathcal{I}}(x) >$ for which $i_{\mathcal{I}}(x) = i_{\mathcal{I}}(x)^2 \bmod (1-x^\nu)$ [67, Ch. 8, §2]. The idempotent can be easily found with *Bezout's algorithm*. Since the generator polynomial $g_{\mathcal{I}}(x)$ and check polynomial $h_{\mathcal{I}}(x)$ are relatively prime, there exist two polynomials $u(x)$ and $v(x)$ such that

$$u(x)g_{\mathcal{I}}(x) + v(x)h_{\mathcal{I}}(x) = 1\,. \tag{5.1}$$

With $i_{\mathcal{I}}(x) = u(x)g_{\mathcal{I}}(x)$, it is easily verified that

$$\begin{aligned} i_{\mathcal{I}}(x)^2 &= i_{\mathcal{I}}(x)(u(x)g_{\mathcal{I}}(x)) \\ &= i_{\mathcal{I}}(x)\underbrace{(u(x)g_{\mathcal{I}}(x) + v(x)h_{\mathcal{I}}(x))}_{1} \\ &= i_{\mathcal{I}}(x)\,. \end{aligned}$$

The term $v(x)h_{\mathcal{I}}(x)$ can always be added because

$$i_{\mathcal{I}}(x)v(x)h_{\mathcal{I}}(x) = u(x)g_{\mathcal{I}}(x)v(x)h_{\mathcal{I}}(x) = 0\,.$$

Each ν-th root of unity α is a zero of $g_{\mathcal{I}}(x)$ or $h_{\mathcal{I}}(x)$. Because $u(x)$ and $h_{\mathcal{I}}(x)$ are relatively prime (Bezout), each zero of $u(x)$ is also a zero of $g_{\mathcal{I}}(x)$. Hence, $u(x)$ does not introduce any new zeros and consequently, $i_{\mathcal{I}}(x)$ generates the same ideal as $g_{\mathcal{I}}(x)$:

$$\mathcal{I}[\nu,\kappa] = < i_{\mathcal{I}}(x) > \ .$$

Note that $u(x)$ can have possible zeros that do not belong to the multiplicative subgroup generated by α.

Definition 5.1 (Minimal ideal) A minimal ideal is an ideal $\mathcal{I}[\nu,\kappa]$ that does not consist of a smaller non-zero ideal. It is denoted with $\mathcal{M}[\nu,\kappa]$. □

Hence, the non-zeros of a minimal ideal are $\{\alpha^i | i \in \mathscr{C}_{q,\nu}^{(s)}\}$, corresponding to a single cyclotomic coset (see Definition 2.8): $\overline{Z}_{\mathcal{M}} = \mathscr{C}_{q,\nu}^{(s)}$. The *idempotent* of a minimal ideal is called a *primitive idempotent*, commonly denoted with $\theta^{(s)}(x)$. Primitive idempotents have a number of interesting properties:

- $$\sum_s \theta^{(s)}(x) = 1$$
- $$\theta^{(i)}(x)\theta^{(j)}(x) = 0 \text{ if } i \neq j$$
- Any ideal can be written as a *direct sum* of minimal ideals:

$$\mathcal{I}[\nu,\kappa] = \oplus_s <\theta^{(s)}(x)> = <\sum_s \theta^{(s)}(x)> \ . \tag{5.2}$$

The following property of cyclic codes turns out to be tightly connected with the filter bank representation of certain cyclic codes.

Theorem 5.2 *There exists an isomorphism Φ between the minimal ideal $\mathcal{M}[\nu,\kappa] =< \theta^{(s)}(x) >$ over $\mathbb{F}_q$ and the Galois field $\mathbb{F}_{q^\kappa}$, defined as follows:*

$$y(x) \in \mathcal{M}[\nu,\kappa] \leftrightarrow y(\xi) \in \mathbb{F}_{q^\kappa}$$

with $\xi \in \overline{Z}_{\mathcal{M}}$. The reverse mapping Φ^{-1} gives a codeword $y(x) \in \mathcal{M}[\nu,\kappa]$ starting from an arbitrary $\gamma^k \in \mathbb{F}_{q^\kappa}$.

$$y[j] = Tr_n\left(\gamma^k \xi^{-j}\right) \ . \tag{5.3}$$

Example 5.1 Continuing Example 2.5, the idempotent

$$i_{\mathcal{I}}(x) = \alpha^4 x + x^2 + \alpha^4 x^3 + x^4 + x^5 + x^6 + \alpha^4 x^7 + x^8 + \alpha^4 x^9 \, . \tag{5.4}$$

This idempotent can be written as a sum of primitive idempotents:

$$i_{\mathcal{I}}(x) = \theta^{(0)}(x) + \theta^{(2)}(x) + \theta^{(4)}(x) + \theta^{(5)}(x) \, . \tag{5.5}$$

All primitive idempotents corresponding to $\mathscr{C}_{9,10}^{(s)}$, $s = 0:5$ are

$$\begin{aligned}
\theta^{(0)}(x) &= 1 + x + x^2 + x^3 + x^4 + x^5 + x^6 + x^7 + x^8 + x^9 \\
\theta^{(1)}(x) &= \alpha^4 + \alpha^3 x + \alpha^5 x^2 + \alpha^1 x^3 + \alpha^7 x^4 + x^5 + \alpha^7 x^6 + \alpha^1 x^7 + \alpha^5 x^8 + \alpha^3 x^9
\end{aligned}$$

$$\theta^{(2)}(x) = \alpha^4 + \alpha^5 x + \alpha^7 x^2 + \alpha^7 x^3 + \alpha^5 x^4 + \alpha^4 x^5 + \alpha^5 x^6 + \alpha^7 x^7 + \alpha^7 x^8 + \alpha^5 x^9$$
$$\theta^{(3)}(x) = \alpha^4 + \alpha^1 x + \alpha^7 x^2 + \alpha^3 x^3 + \alpha^5 x^4 + x^5 + \alpha^5 x^6 + \alpha^3 x^7 + \alpha^7 x^8 + \alpha^1 x^9$$
$$\theta^{(4)}(x) = \alpha^4 + \alpha^7 x + \alpha^5 x^2 + \alpha^5 x^3 + \alpha^7 x^4 + \alpha^4 x^5 + \alpha^7 x^6 + \alpha^5 x^7 + \alpha^5 x^8 + \alpha^7 x^9$$
$$\theta^{(5)}(x) = 1 + \alpha^4 x + x^2 + \alpha^4 x^3 + x^4 + \alpha^4 x^5 + x^6 + \alpha^4 x^7 + x^8 + \alpha^4 x^9$$

The ideal $\mathcal{I}[10,6]$ can also be represented as a direct sum of minimal ideals:

$$\mathcal{I}[10,6] = <\theta^{(0)}(x)> \oplus <\theta^{(2)}(x)> \oplus <\theta^{(4)}(x)> \oplus <\theta^{(5)}(x)> \ .$$

Consequently, this code is also generated by the following generator matrix:

$$\mathbf{G}_{\mathcal{I}}^{V} = \left[\begin{array}{c|cc|cc|c} 1 & \alpha^4 & 0 & \alpha^4 & 0 & \alpha^4 \\ 1 & \alpha^1 & \alpha^4 & \alpha^3 & \alpha^4 & 1 \\ 1 & \alpha^5 & \alpha^1 & \alpha^7 & \alpha^3 & \alpha^4 \\ 1 & 1 & \alpha^5 & 1 & \alpha^7 & 1 \\ 1 & 0 & 1 & 0 & 1 & \alpha^4 \\ 1 & \alpha^4 & 0 & \alpha^4 & 0 & 1 \\ 1 & \alpha^1 & \alpha^4 & \alpha^3 & \alpha^4 & \alpha^4 \\ 1 & \alpha^5 & \alpha^1 & \alpha^7 & \alpha^3 & 1 \\ 1 & 1 & \alpha^5 & 1 & \alpha^7 & \alpha^4 \\ 1 & 0 & 1 & 0 & 1 & 1 \end{array}\right] .$$

Consider the minimal ideal $\mathcal{M}[10,2] = <\theta^{(1)}(x)>$. The isomorphism between all codewords of $\mathcal{M}[10,2]$ (except the all-zero codeword) and the elements of $\mathbb{F}_{3^4}$ is shown in Table 5.1. △

5.3 2-D Cyclic Codes

The concept of a 2-D cyclic code can be illustrated by means of a codeword $y(x) \in \mathcal{R}[\nu, \kappa]$. This codeword is rewritten as follows, similar to Eq. 3.4.1[1]

$$y(x) = \left[x^{\nu'(0:M-1)}\right] \mathbf{\Gamma} \left[x^{0:\nu'-1}\right]^T \bmod (1 - x^{\nu}) .$$

Imagine $y(x)$ is written as a bivariate polynomial in v, w with $x = w$ and $x^{\nu'} = v$. (Or $x^t = v^s w^{s'}$ iff. $t = Ms + s'$):

$$y(v,w) = \left[v^{0:M-1}\right] \mathbf{\Gamma} \left[w^{0:\nu'-1}\right]^T \bmod (1 - x^{\nu}) . \tag{5.6}$$

Next, it is seen that the code $\mathcal{R}[\nu, \kappa]$ is 2-D cyclic, i.e. cyclic in v *and* in w:

$$y(v,w) \in \mathcal{R}[\nu,\kappa] \implies vy(v,w) \in \mathcal{R}[\nu,\kappa] \bmod (1 - v^M)$$

[1]Note that the size of the matrix $\mathbf{\Gamma}$ in Eq. 3.4.1 is $M \times K$. Here, $\nu' - K$ zero columns have been added to obtain the correct size $M \times \nu'$.

Table 5.1: Isomorphism Φ between codewords of $\mathcal{M}[10,2]$ and $\mathbb{F}_{3^4}$.

$\Phi y(x)$	$y(x)$	$\Phi y(x)$	$y(x)$
γ^{65}	$0\ \beta^4\ \beta^7\ \beta^7\ \beta^4\ 0\ 1\ \beta^3\ \beta^3\ 1$	γ^{49}	$\beta^7\ \beta^7\ \beta^4\ 0\ 1\ \beta^3\ \beta^3\ 1\ 0\ \beta^4$
γ^{75}	$0\ \beta^5\ 1\ 1\ \beta^5\ 0\ \beta^1\ \beta^4\ \beta^4\ \beta^1$	γ^{4}	$\beta^7\ \beta^3\ \beta^1\ \beta^2\ \beta^1\ \beta^3\ \beta^7\ \beta^5\ \beta^6\ \beta^5$
γ^{5}	$0\ \beta^6\ \beta^1\ \beta^1\ \beta^6\ 0\ \beta^2\ \beta^5\ \beta^5\ \beta^2$	γ^{30}	$\beta^7\ \beta^6\ 1\ \beta^4\ \beta^2\ \beta^3\ \beta^2\ \beta^4\ 1\ \beta^6$
γ^{15}	$0\ \beta^7\ \beta^2\ \beta^2\ \beta^7\ 0\ \beta^3\ \beta^6\ \beta^6\ \beta^3$	γ^{41}	$\beta^7\ \beta^4\ 0\ 1\ \beta^3\ \beta^3\ 1\ 0\ \beta^4\ \beta^7$
γ^{25}	$0\ 1\ \beta^3\ \beta^3\ 1\ 0\ \beta^4\ \beta^7\ \beta^7\ \beta^4$	γ^{17}	$1\ \beta^3\ \beta^3\ 1\ 0\ \beta^4\ \beta^7\ \beta^7\ \beta^4\ 0$
γ^{35}	$0\ \beta^1\ \beta^4\ \beta^4\ \beta^1\ 0\ \beta^5\ 1\ 1\ \beta^5$	γ^{51}	$1\ \beta^5\ 0\ \beta^1\ \beta^4\ \beta^4\ \beta^1\ 0\ \beta^5\ 1$
γ^{45}	$0\ \beta^2\ \beta^5\ \beta^5\ \beta^2\ 0\ \beta^6\ \beta^1\ \beta^1\ \beta^6$	γ^{42}	$1\ \beta^2\ \beta^6\ \beta^4\ \beta^5\ \beta^4\ \beta^6\ \beta^2\ 1\ \beta^1$
γ^{55}	$0\ \beta^3\ \beta^6\ \beta^6\ \beta^3\ 0\ \beta^7\ \beta^2\ \beta^2\ \beta^7$	γ^{58}	$1\ \beta^1\ 1\ \beta^2\ \beta^6\ \beta^4\ \beta^5\ \beta^4\ \beta^6\ \beta^2$
γ^{57}	$\beta^4\ \beta^7\ \beta^7\ \beta^4\ 0\ 1\ \beta^3\ \beta^3\ 1\ 0$	γ^{73}	$1\ 0\ \beta^4\ \beta^7\ \beta^7\ \beta^4\ 0\ 1\ \beta^3\ \beta^3$
γ^{6}	$\beta^4\ \beta^2\ \beta^3\ \beta^2\ \beta^4\ 1\ \beta^6\ \beta^7\ \beta^6\ 1$	γ^{46}	$1\ \beta^6\ \beta^7\ \beta^6\ 1\ \beta^4\ \beta^2\ \beta^3\ \beta^2\ \beta^4$
γ^{19}	$\beta^4\ \beta^4\ \beta^1\ 0\ \beta^5\ 1\ 1\ \beta^5\ 0\ \beta^1$	γ^{59}	$1\ 1\ \beta^5\ 0\ \beta^1\ \beta^4\ \beta^4\ \beta^1\ 0\ \beta^5$
γ^{54}	$\beta^4\ 1\ \beta^6\ \beta^7\ \beta^6\ 1\ \beta^4\ \beta^2\ \beta^3\ \beta^2$	γ^{14}	$1\ \beta^4\ \beta^2\ \beta^3\ \beta^2\ \beta^4\ 1\ \beta^6\ \beta^7\ \beta^6$
1	$\beta^4\ \beta^3\ \beta^5\ \beta^1\ \beta^7\ 1\ \beta^7\ \beta^1\ \beta^5\ \beta^3$	γ^{40}	$1\ \beta^7\ \beta^1\ \beta^5\ \beta^3\ \beta^4\ \beta^3\ \beta^5\ \beta^1\ \beta^7$
γ^{11}	$\beta^4\ \beta^1\ 0\ \beta^5\ 1\ 1\ \beta^5\ 0\ \beta^1\ \beta^4$	γ^{27}	$\beta^1\ \beta^4\ \beta^4\ \beta^1\ 0\ \beta^5\ 1\ 1\ \beta^5\ 0$
γ^{2}	$\beta^4\ \beta^6\ \beta^2\ 1\ \beta^1\ 1\ \beta^2\ \beta^6\ \beta^4\ \beta^5$	γ^{50}	$\beta^1\ 1\ \beta^2\ \beta^6\ \beta^4\ \beta^5\ \beta^4\ \beta^6\ \beta^2\ 1$
γ^{18}	$\beta^4\ \beta^5\ \beta^4\ \beta^6\ \beta^2\ 1\ \beta^1\ 1\ \beta^2\ \beta^6$	γ^{61}	$\beta^1\ \beta^6\ 0\ \beta^2\ \beta^5\ \beta^5\ \beta^2\ 0\ \beta^6\ \beta^1$
γ^{33}	$\beta^4\ 0\ 1\ \beta^3\ \beta^3\ 1\ 0\ \beta^4\ \beta^7\ \beta^7$	γ^{52}	$\beta^1\ \beta^3\ \beta^7\ \beta^5\ \beta^6\ \beta^5\ \beta^7\ \beta^3\ \beta^1\ \beta^2$
γ^{67}	$\beta^5\ 1\ 1\ \beta^5\ 0\ \beta^1\ \beta^4\ \beta^4\ \beta^1\ 0$	γ^{68}	$\beta^1\ \beta^2\ \beta^1\ \beta^3\ \beta^7\ \beta^5\ \beta^6\ \beta^5\ \beta^7\ \beta^3$
γ^{43}	$\beta^5\ 0\ \beta^1\ \beta^4\ \beta^4\ \beta^1\ 0\ \beta^5\ 1\ 1$	γ^{3}	$\beta^1\ 0\ \beta^5\ 1\ 1\ \beta^5\ 0\ \beta^1\ \beta^4\ \beta^4$
γ^{16}	$\beta^5\ \beta^3\ \beta^4\ \beta^3\ \beta^5\ \beta^1\ \beta^7\ 1\ \beta^7\ \beta^1$	γ^{56}	$\beta^1\ \beta^7\ 1\ \beta^7\ \beta^1\ \beta^5\ \beta^3\ \beta^4\ \beta^3\ \beta^5$
γ^{29}	$\beta^5\ \beta^5\ \beta^2\ 0\ \beta^6\ \beta^1\ \beta^1\ \beta^6\ 0\ \beta^2$	γ^{69}	$\beta^1\ \beta^1\ \beta^6\ 0\ \beta^2\ \beta^5\ \beta^5\ \beta^2\ 0\ \beta^6$
γ^{64}	$\beta^5\ \beta^1\ \beta^7\ 1\ \beta^7\ \beta^1\ \beta^5\ \beta^3\ \beta^4\ \beta^3$	γ^{24}	$\beta^1\ \beta^5\ \beta^3\ \beta^4\ \beta^3\ \beta^5\ \beta^1\ \beta^7\ 1\ \beta^7$
γ^{10}	$\beta^5\ \beta^4\ \beta^6\ \beta^2\ 1\ \beta^1\ 1\ \beta^2\ \beta^6\ \beta^4$	γ^{37}	$\beta^2\ \beta^5\ \beta^5\ \beta^2\ 0\ \beta^6\ \beta^1\ \beta^1\ \beta^6\ 0$
γ^{21}	$\beta^5\ \beta^2\ 0\ \beta^6\ \beta^1\ \beta^1\ \beta^6\ 0\ \beta^2\ \beta^5$	γ^{34}	$\beta^2\ \beta^6\ \beta^4\ \beta^5\ \beta^4\ \beta^6\ \beta^2\ 1\ \beta^1\ 1$
γ^{12}	$\beta^5\ \beta^7\ \beta^3\ \beta^1\ \beta^2\ \beta^1\ \beta^3\ \beta^7\ \beta^5\ \beta^6$	γ^{60}	$\beta^2\ \beta^1\ \beta^3\ \beta^7\ \beta^5\ \beta^6\ \beta^5\ \beta^7\ \beta^3\ \beta^1$
γ^{28}	$\beta^5\ \beta^6\ \beta^5\ \beta^7\ \beta^3\ \beta^1\ \beta^2\ \beta^1\ \beta^3\ \beta^7$	γ^{71}	$\beta^2\ \beta^7\ 0\ \beta^3\ \beta^6\ \beta^6\ \beta^3\ 0\ \beta^7\ \beta^2$
γ^{77}	$\beta^6\ \beta^1\ \beta^1\ \beta^6\ 0\ \beta^2\ \beta^5\ \beta^5\ \beta^2\ 0$	γ^{62}	$\beta^2\ \beta^4\ 1\ \beta^6\ \beta^7\ \beta^6\ 1\ \beta^4\ \beta^2\ \beta^3$
γ^{38}	$\beta^6\ \beta^7\ \beta^6\ 1\ \beta^4\ \beta^2\ \beta^3\ \beta^2\ \beta^4\ 1$	γ^{78}	$\beta^2\ \beta^3\ \beta^2\ \beta^4\ 1\ \beta^6\ \beta^7\ \beta^6\ 1\ \beta^4$
γ^{53}	$\beta^6\ 0\ \beta^2\ \beta^5\ \beta^5\ \beta^2\ 0\ \beta^6\ \beta^1\ \beta^1$	γ^{13}	$\beta^2\ 0\ \beta^6\ \beta^1\ \beta^1\ \beta^6\ 0\ \beta^2\ \beta^5\ \beta^5$
γ^{26}	$\beta^6\ \beta^4\ \beta^5\ \beta^4\ \beta^6\ \beta^2\ 1\ \beta^1\ 1\ \beta^2$	γ^{66}	$\beta^2\ 1\ \beta^1\ 1\ \beta^2\ \beta^6\ \beta^4\ \beta^5\ \beta^4\ \beta^6$
γ^{39}	$\beta^6\ \beta^6\ \beta^3\ 0\ \beta^7\ \beta^2\ \beta^2\ \beta^7\ 0\ \beta^3$	γ^{79}	$\beta^2\ \beta^2\ \beta^7\ 0\ \beta^3\ \beta^6\ \beta^6\ \beta^3\ 0\ \beta^7$
γ^{74}	$\beta^6\ \beta^2\ 1\ \beta^1\ 1\ \beta^2\ \beta^6\ \beta^4\ \beta^5\ \beta^4$	γ^{47}	$\beta^3\ \beta^6\ \beta^6\ \beta^3\ 0\ \beta^7\ \beta^2\ \beta^2\ \beta^7\ 0$
γ^{20}	$\beta^6\ \beta^5\ \beta^7\ \beta^3\ \beta^1\ \beta^2\ \beta^1\ \beta^3\ \beta^7\ \beta^5$	γ^{9}	$\beta^3\ \beta^3\ 1\ 0\ \beta^4\ \beta^7\ \beta^7\ \beta^4\ 0\ 1$
γ^{31}	$\beta^6\ \beta^3\ 0\ \beta^7\ \beta^2\ \beta^2\ \beta^7\ 0\ \beta^3\ \beta^6$	γ^{44}	$\beta^3\ \beta^7\ \beta^5\ \beta^6\ \beta^5\ \beta^7\ \beta^3\ \beta^1\ \beta^2\ \beta^1$
γ^{22}	$\beta^6\ 1\ \beta^4\ \beta^2\ \beta^3\ \beta^2\ \beta^4\ 1\ \beta^6\ \beta^7$	γ^{70}	$\beta^3\ \beta^2\ \beta^4\ 1\ \beta^6\ \beta^7\ \beta^6\ 1\ \beta^4\ \beta^2$
γ^{7}	$\beta^7\ \beta^2\ \beta^2\ \beta^7\ 0\ \beta^3\ \beta^6\ \beta^6\ \beta^3\ 0$	γ^{1}	$\beta^3\ 1\ 0\ \beta^4\ \beta^7\ \beta^7\ \beta^4\ 0\ 1\ \beta^3$
γ^{32}	$\beta^7\ \beta^1\ \beta^5\ \beta^3\ \beta^4\ \beta^3\ \beta^5\ \beta^1\ \beta^7\ 1$	γ^{72}	$\beta^3\ \beta^5\ \beta^1\ \beta^7\ 1\ \beta^7\ \beta^1\ \beta^5\ \beta^3\ \beta^4$
γ^{48}	$\beta^7\ 1\ \beta^7\ \beta^1\ \beta^5\ \beta^3\ \beta^4\ \beta^3\ \beta^5\ \beta^1$	γ^{8}	$\beta^3\ \beta^4\ \beta^3\ \beta^5\ \beta^1\ \beta^7\ 1\ \beta^7\ \beta^1\ \beta^5$
γ^{63}	$\beta^7\ 0\ \beta^3\ \beta^6\ \beta^6\ \beta^3\ 0\ \beta^7\ \beta^2\ \beta^2$	γ^{23}	$\beta^3\ 0\ \beta^7\ \beta^2\ \beta^2\ \beta^7\ 0\ \beta^3\ \beta^6\ \beta^6$
γ^{36}	$\beta^7\ \beta^5\ \beta^6\ \beta^5\ \beta^7\ \beta^3\ \beta^1\ \beta^2\ \beta^1\ \beta^3$	γ^{76}	$\beta^3\ \beta^1\ \beta^2\ \beta^1\ \beta^3\ \beta^7\ \beta^5\ \beta^6\ \beta^5\ \beta^7$

$$y(v,w) \in \mathcal{R}\left[\nu,\kappa\right] \quad \Longrightarrow \quad wy(v,w) \in \mathcal{R}\left[\nu,\kappa\right] \bmod \left(1-w^{\nu'}\right).$$

Starting from the bivariate polynomial $y(v,w)$, the interleaved codeword $y^{\Pi}(x) \in \mathcal{R}^{\Pi_{\nu'}}\left[\nu,\kappa\right]$ can be obtained by assigning different powers of x to v and w, namely $w = x^M$ and $v = x$ (see again Section 3.4.2). The codeword of the RS code in its filter bank form $y^{\Pi}(x)$ becomes:

$$y^{\Pi}(x) \quad = \quad \left[x^{0:M-1}\right] \mathbf{\Gamma} \left[x^{(0:\nu'-1)M}\right]^T \bmod (1-x^{\nu}).$$

Again, the 2-D cyclic character is obvious.

Example 5.2 This all can be clarified with the following example. Let the first column of the block Toeplitz generator matrix $\mathbf{G}^{V}_{\mathcal{R}^{\Pi_{\nu'}}}$ for $\mathcal{R}\left[15,10\right]$ with $M = 3$ (See Example 4.4) serve as codeword

$$y^{\Pi}(x) = \alpha^{14} + \alpha^{5}x^{1} + \alpha^{6}x^{2} + \alpha^{4}x^{3} + \alpha^{6}x^{4} + 1x^{7} + \alpha^{5}x^{8}\,.$$

The non-interleaved version

$$y(x) = \alpha^{14} + \alpha^{4}x^{1} + \alpha^{5}x^{5} + \alpha^{6}x^{6} + x^{7} + \alpha^{6}x^{10} + \alpha^{5}x^{12}$$

is a codeword of $\mathcal{R}\left[15,10\right]$ and can be written as a bivariate polynomial,

$$y(v,w) = \begin{bmatrix}1 & v & v^2\end{bmatrix} \underbrace{\begin{bmatrix}\alpha^{14} & \alpha^{4} & 0 & 0 & 0\\ \alpha^{5} & \alpha^{6} & 1 & 0 & 0\\ \alpha^{6} & 0 & \alpha^{5} & 0 & 0\end{bmatrix}}_{\mathbf{\Gamma}} \begin{bmatrix}1\\ w\\ w^2\\ w^3\\ w^4\end{bmatrix}.$$

It can be verified that the cyclic shifts in v and w belong to the RS code:

$$vy(v,w) \bmod \left(1-v^{M}\right) \quad = \quad x^{5}y(x) \bmod \left(1-x^{\nu}\right) \in \mathcal{R}\left[15,10\right] \qquad (5.7)$$

$$wy(v,w) \bmod \left(1-w^{\nu'}\right) \quad = \quad xy(x) \bmod \left(1-x^{\nu}\right) \in \mathcal{R}\left[15,10\right]. \qquad (5.8)$$

Note that if the matrix $\mathbf{\Gamma}$ is read out column-wise, the codeword $\mathbf{y}^{\Pi}$ is obtained. If it is read in a row-wise fashion, $\mathbf{y}$ is obtained. △

5.3.1 Type I Filter Bank Decomposition of RS Codes

The interleaver operation between $y(x)$ and $y^{\Pi}(x)$, generated by the filter bank, leads to the following key decomposition (Eq. 3.20):

$$\boxed{\mathbf{G}_{\mathcal{R}} = \mathbf{\Pi}\underbrace{\mathbf{G}_{\mathcal{C}}\mathbf{G}_{\mathcal{D}}\mathbf{G}_{\mathcal{A}}}_{\mathbf{G}^{V}_{\mathcal{R}^{\Pi_{\nu'}}}}\mathbf{V}}$$

In order to make a clear distinction with a similar decomposition presented below, we will refer to this one as a decomposition of type I. Now, the subband codes are further studied. In the construction in Section 4.3, a ν-point Cook-Toom algorithm is applied to decompose the Toeplitz generator matrix $\mathbf{G}_{\mathcal{R}}$, resulting in a diagonal matrix $\mathbf{\Delta}$. The next step transforms this diagonal matrix back into M Toeplitz matrices, which then correspond to the subband filters. At that point, the roots of the RS code are distributed 'evenly' over the Toeplitz matrices, and thus over the subband matrices. The following theorem states more accurately how this distribution takes place:

Theorem 5.3 *Consider an RS code $\mathcal{R}[\nu, \kappa]$ with roots $\{\alpha^b, \alpha^{b+1}, ..., \alpha^{b+\nu-\kappa-1}\}$ and the corresponding filter bank as defined in Theorem 4.2. A root α^p will be assigned to band m if and only if $m = p - \tilde{b} \bmod M$.*

Proof : This follows from the construction in Section 4.3. □

Example 5.3 In the $\mathcal{R}[15, 10]$ example, the roots are $\{\alpha^3, \alpha^4, \alpha^5, \alpha^6, \alpha^7\}$. According to the previous theorem, α^5 is a root of the subband filter in band 0, while α^3, α^6 respectively α^4, α^7 are roots in the bands 1 and 2. △

Note that the subband code with roots belonging to the multiplicative subgroup $\alpha^0, \alpha^M, \ldots, \alpha^{(\nu'-1)M}$ is a non-primitive BCH code of length ν', which is *cyclic*. The other subband codes are *not* cyclic. However, the next section shows that *all* subband codes can be made cyclic if the interleaver is changed accordingly. This is an interesting structure which allows us to generalize filter bank techniques to other codes. It is also the foundation of unequal error protecting codes, as will be discussed later on.

5.3.2 Type II Filter Bank Decomposition of RS Codes

In order to obtain a so-called type II decomposition with cyclic subband codes, each subband code is transformed into a (cyclic) non-primitive BCH code. Let us start with an illustrative example first:

Example 5.4 Continuing the example with $\mathcal{R}[15, 10]$, the first band ($m = 0$) has a single root $-\alpha^5 = \alpha^5$ ($d_0(x) = \alpha^5 + x$). To obtain a cyclic BCH code, the roots have to belong to the multiplicative subgroup $\alpha^0, \alpha^3, \alpha^6, \alpha^9, \alpha^{12}$. If x is transformed into $\alpha^{-5 \cdot 2} x'$, the subband code becomes $d_0(x') = \alpha^5(x' + 1)$ with root α^0. In general, x is transformed into $\alpha^{-\nu' i} x'$ with i the smallest integer such that $M | i\nu' + m + \tilde{b}$. From a matrix point of view, the generator matrix

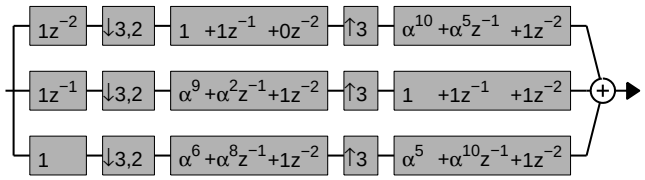


Figure 5.1: Type II critically subsampled filter bank decomposition of $\mathcal{R}\,[15, 10]$ with cyclic subband codes $\mathcal{D}'_m$ and cyclic synthesis bank codes $\mathcal{C}'_m$ as discussed in Example 5.4.

of the transformed subband code is

$$\mathbf{G}_{\mathcal{D}'_0} = \begin{bmatrix} 1 & & & & \\ 1 & 1 & & & \\ & 1 & 1 & & \\ & & 1 & 1 & \\ & & & 1 & 1 \end{bmatrix} = \underbrace{\begin{bmatrix} \alpha^{-10} & & & & \\ & \alpha^{-5} & & & \\ & & 1 & & \\ & & & \alpha^{-10} & \\ & & & & \alpha^{-5} \end{bmatrix}}_{\mathbf{E}_0} \begin{bmatrix} \alpha^5 & & & & \\ 1 & \alpha^5 & & & \\ & 1 & \alpha^5 & & \\ & & 1 & \alpha^5 & \\ & & & 1 & \alpha^5 \end{bmatrix} \begin{bmatrix} \alpha^5 & & & \\ & 1 & & \\ & & \alpha^{10} & \\ & & & \alpha^5 \end{bmatrix}.$$

Note that the matrices that describe the transformations are diagonal matrices whose elements are multiples of α^5.

The roots of the second band already belong to the multiplicative subgroup, so a variable transformation is not necessary ($x = x'$):

$$\mathbf{G}_{\mathcal{D}'_1} = \mathbf{G}_{\mathcal{D}_1} = \begin{bmatrix} \alpha^9 & & & \\ \alpha^2\alpha^9 & & & \\ 1 & \alpha^2\alpha^9 & & \\ & 1 & \alpha^2 & \\ & & 1 & \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix}}_{\mathbf{E}_1} \begin{bmatrix} \alpha^4 & & & \\ \alpha^{12} & \alpha^4 & & \\ \alpha^{10}\alpha^{12} & \alpha^4 & & \\ & \alpha^{10}\alpha^{12} & & \\ & & \alpha^{10} & \end{bmatrix} \begin{bmatrix} \alpha^5 & & \\ & \alpha^5 & \\ & & \alpha^5 \end{bmatrix}.$$

The third band has roots α^4, α^7. With $i = 1$, $M|i\nu' + m + \tilde{b}$ such that x is replaced by $\alpha^{-5}x'$. The resulting BCH code has roots α^9, α^{12} ($d_2(x') = \alpha^6 + \alpha^8 x' + x'^2$). Its generator matrix is

$$\mathbf{G}_{\mathcal{D}'_2} = \begin{bmatrix} \alpha^6 & & & \\ \alpha^8\alpha^6 & & & \\ 1 & \alpha^8\alpha^6 & & \\ & 1 & \alpha^8 & \\ & & 1 & \end{bmatrix} = \underbrace{\begin{bmatrix} \alpha^{-5} & & & & \\ & \alpha^{-10} & & & \\ & & 1 & & \\ & & & \alpha^{-5} & \\ & & & & \alpha^{-10} \end{bmatrix}}_{\mathbf{E}_2} \begin{bmatrix} \alpha^6 & & & \\ \alpha^{13} & \alpha^6 & & \\ \alpha^{10}\alpha^{13} & \alpha^6 & & \\ & \alpha^{10}\alpha^{13} & & \\ & & \alpha^{10} & \end{bmatrix} \begin{bmatrix} \alpha^5 & & \\ & \alpha^{10} & \\ & & 1 \end{bmatrix}.$$

△

With the newly obtained cyclic subband codes, the matrix decomposition of Eq. 3.20 becomes

$$\mathbf{G}_{\mathcal{R}} = \mathbf{\Pi}\mathbf{G}_{\mathcal{C}}\mathbf{E}\mathbf{G}_{\mathcal{D}'}\mathbf{V}'' \tag{5.9}$$

with $\mathbf{E}\left[m + M(0:\nu'-1), m + M(0:\nu'-1)\right] = \mathbf{E}_m$. To obtain a filter bank like decomposition, the matrix $\mathbf{E}$ should be incorporated into a possibly different interleaver and synthesis bank. Using some matrix algebra, it is possible to show that

$$\mathbf{G}_{\mathcal{C}}\mathbf{E} = \mathbf{T}\mathbf{G}_{\mathcal{C}'}. \tag{5.10}$$

such that Eq. 5.9 becomes

$$\mathbf{G}_{\mathcal{R}} = \mathbf{\Pi}\mathbf{T}\mathbf{G}_{\mathcal{C}'}\mathbf{G}_{\mathcal{D}'}\mathbf{V}''\,.$$

In this equation, $\mathbf{G}_{\mathcal{C}'}$ is a block diagonal matrix with equal blocks defining the synthesis bank (see Eq. 4.27):

$$\mathbf{G}_{\mathcal{C}'} = \mathbf{I}_{\nu'} \otimes \tilde{\mathbf{F}'}_M^{-1}\,. \tag{5.11}$$

$\mathbf{T}$ is also a block diagonal matrix, where the j-th block is a rotated interleaver matrix:

$$\mathbf{T}\left[jM + (0:M-1), jM + (0:M-1)\right] = \mathbf{\Pi}_{\nu' mod M}\mathbf{R}_M^{-j}\,. \tag{5.12}$$

Hence, it is seen that $\tilde{\mathbf{F}'}_M^{-1}$ is identical to the original $\tilde{\mathbf{F}}_M^{-1}$, but with the rows interleaved. As a consequence, the synthesis bank of the type II filter bank decomposition contains the same filters, but in different bands. Also note that $\mathbf{T}$ itself can be seen as one big interleaver matrix, such that we define an interleaver matrix

$$\begin{aligned}\mathbf{\Pi}' &= \mathbf{\Pi}\mathbf{T} \\ &= \mathbf{I}_{\nu}\left[(0:\nu-1 \bmod M) + M(0:\nu-1 \bmod \nu')\right]\,.\end{aligned}$$

Finally, the type II matrix decomposition of a RS code generator matrix is obtained:

$$\boxed{\mathbf{G}_{\mathcal{R}} = \mathbf{\Pi}'\mathbf{G}_{\mathcal{C}'}\mathbf{G}_{\mathcal{D}'}\mathbf{V}}. \tag{5.13}$$

In the rest of this section, another more insightful approach is given, that explains why the above matrix decomposition represents an RS code. This exposition is based on the theory of 2-D cyclic codes and more specifically on Berlekamp's seminal paper [5]. In this paper, a concatenation (in the sense of Forney) of two cyclic codes is considered. This means that an outer code $\mathcal{I}_{\text{outer}}$ over $\mathbb{F}_{2^{\kappa_{\text{inner}}}}$ is concatenated with a small binary inner code $\mathcal{I}_{\text{inner}}[M, \kappa_{\text{inner}}]$. Due to the one-to-one mapping (isomorphism Φ) between the codewords of the inner code and the elements of $\mathbb{F}_{2^{\kappa_{\text{inner}}}}$, each symbol of the outer codeword is mapped onto M binary codeword symbols of the inner code (see Figure 5.2).

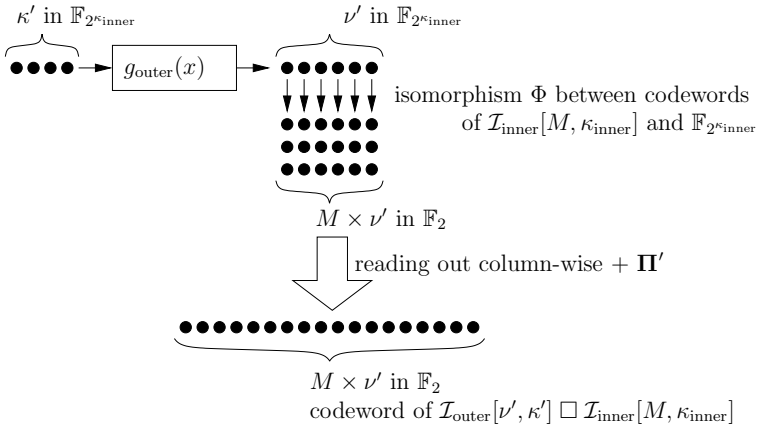


Figure 5.2: Concatenation (in the sense of Forney) of two cyclic codes.

The resulting code $\mathcal{I}_{\text{outer}}[\nu', \kappa'] \square \mathcal{I}_{\text{inner}}[M, \kappa_{\text{inner}}]$ can best be described as a 2-D code. For example, let $y(w)$ be a codeword of the outer code with coefficients in $\mathbb{F}_{2^{\kappa_{\text{inner}}}}$. Each coefficient is then mapped onto a binary codeword (polynomial in v of length ν) such that the overall codeword is a bivariate polynomial $y(v, w)$.

Next, it is shown how this theory of concatenated codes fits the filter bank representation. Focus on a specific band m, with an outer (subband) code ${\mathcal{D}'}_m(\nu', \kappa'_m)$ concatenated with an inner code ${\mathcal{C}'}_m(M, 1)$. As defined in Eq. 4.16, κ'_m is used to denote the dimension of the code ${\mathcal{D}'}_m(\nu', \kappa'_m)$. The concatenated code is denoted with ${\mathcal{D}'}_m(\nu', \kappa'_m) \square {\mathcal{C}'}_m(M, 1)$. Theorem 5.2 states that an isomorphism Φ exists between the codewords of the inner code ${\mathcal{C}'}_m(M, 1)$ and $\mathbb{F}_q$. The inverse mapping is denoted with Φ^{-1}. This isomorphism is called upon to map each codeword symbol of the outer code onto a codeword of the inner code. For example, assume $y(w)$ is a codeword of the outer code, represented with a polynomial of length ν':

$$y(w) \in {\mathcal{D}'}_m(\nu', \kappa'_m) \,.$$

Each coefficient of $y(w)$ is mapped (using Φ^{-1}) onto a codeword of the inner code, which is represented by a polynomial in v. A codeword of the concatenated code can consequently be represented with a bivariate polynomial

$$y(v, w) = \sum_{n=0}^{\nu'-1} \Phi^{-1}(y\,[n]) w^n \,.$$

Let us now apply an extension of the results of Berlekamp and Justesen in [5] to a specific band m of the filter bank.

Theorem 5.4 *The code obtained by a concatenation (in the sense of Forney) of two* cyclic *codes* ${\mathcal{D}'}_m(\nu', \kappa'_m)$ *and* ${\mathcal{C}'}_m(M, 1)$ *is cyclic if the inner code is a minimal ideal*

Proof : Let $\alpha^{\nu' s}$ be a single nonzero of ${\mathcal{C}'}_m(M, 1)$. Subsequently, we prove that ${\mathcal{D}'}_m(\nu', \kappa'_m) \square {\mathcal{C}'}_m(M, 1)$ is cyclic. It is clear that

$$wy(v, w) \in {\mathcal{D}'}_m(\nu', \kappa'_m) \square {\mathcal{C}'}_m(M, 1)$$

since $wy(w) \bmod (1 - w^{\nu'}) \in {\mathcal{D}'}_m(\nu', \kappa'_m)$. Secondly, a codeword $\alpha^{\nu' s} y(w) \in {\mathcal{D}'}_m(\nu', \kappa'_m)$ is mapped onto $vy(w) \bmod (1 - v^M)$. Hence,

$$vwy(v, w) \bmod (1 - v^M)(1 - w^{\nu'}) \in {\mathcal{D}'}_m(\nu', \kappa'_m) \square {\mathcal{C}'}_m(M, 1).$$

Hence, the concatenated code is 2-D cyclic. The transformation $x = vw$ transforms the bivariate polynomial $y(v, w)$ into a univariate polynomial $y(x)$, such that the concatenated code is indeed cyclic. In general, $v^s w^{s'}$ corresponds with x^t if $t = s \bmod M$ and $t = s' \bmod \nu'$. This transformation is exactly given by the interleaver $\mathbf{\Pi}'$. □

Finally, it is also possible to tell something about the roots of the concatenated code.

Theorem 5.5 *Let* $\alpha^{Ms'}$ *be a nonzero of the outer cyclic subband code* ${\mathcal{D}'}_m(\nu', \kappa'_m)$. *With* $\alpha^{\nu' s}$ *a nonzero of the inner cyclic code* ${\mathcal{C}'}_m(M, 1)$, $\alpha^{(M+\nu')t}$ *(of the synthesis bank) is a nonzero of the concatenated code* ${\mathcal{D}'}_m(\nu', \kappa'_m) \square {\mathcal{C}'}_m(M, 1)$ *if* $t = s \bmod M$ *and* $t = s' \bmod \nu'$.

Proof : Indeed,

$$\begin{aligned} y(\alpha^{Mt} \alpha^{\nu' t}) &= y(\alpha^{\nu' s}, \alpha^{Ms'}) \\ &= \sum_{n=0}^{\nu'-1} y_{{\mathcal{C}'}_m}(\alpha^{\nu' s}) \alpha^{ns'M} \\ &= \sum_{n=0}^{\nu'-1} y\,[n]\, \alpha^{ns'M} \\ &\neq 0 \end{aligned}$$

□

Using the fact that any ideal can be written as a direct sum of minimal ideals (Eq. 5.2), all non-zeros of the code $\mathcal{R}\,[\nu, \kappa]$ can be realized as a direct sum of concatenated codes:

$$\mathcal{R}\,[\nu, \kappa] = \oplus_{m=0}^{M-1} {\mathcal{D}'}_m(\nu', \kappa'_m) \square {\mathcal{C}'}_m(M, 1)\,.$$

With this representation, each band is responsible for realizing part of the nonzeros of the overall RS code. Let us give an example to explain this more in detail.

Example 5.5 As an example, consider the first band ($m = 0$) of the filter bank discussed in Example 5.4 (Figure 5.1). The inner code $\mathcal{C}'_0(3,1)$ is a minimal ideal since it has a single nonzero α^5. As shown in Section 5.2, a minimal ideal is generated by a unique primitive idempotent $\theta^{(1)}(v) = 1 + \alpha^{10}v + \alpha^5 v^2$. Table 5.2 shows the isomorphism Φ between the codewords of $\mathcal{C}'_0(3,1)$ and $\mathbb{F}_{2^4}$. Note that the primitive idempotent is mapped onto the identity element: $\Phi(\theta^{(1)}(v)) = \alpha^0$, which is always the case in fields of binary characteristic. The inverse mapping Φ^{-1} from $\mathbb{F}_{2^4}$ to (the codewords of) $\mathcal{C}'_0(3,1)$ is defined using the trace (See Eq. 5.3).

Table 5.2: Isomorphism Φ between codewords of the inner code, which is a minimal ideal, and $\mathbb{F}_{2^4}$. The idempotent $i_1(v)$ is underlined.

dataword	codeword $y_{\mathcal{C}'_m}(v)$	$\Phi y_{\mathcal{C}'_m}(v) \in \mathbb{F}_{2^4}$
0	0	0
1	$\alpha^{10} + \alpha^5 v + v^2$	α^{10}
α^1	$\alpha^{11} + \alpha^6 v + \alpha^1 v^2$	α^{11}
α^2	$\alpha^{12} + \alpha^7 v + \alpha^2 v^2$	α^{12}
α^3	$\alpha^{13} + \alpha^8 v + \alpha^3 v^2$	α^{13}
α^4	$\alpha^{14} + \alpha^9 v + \alpha^4 v^2$	α^{14}
$\underline{\alpha^5}$	$\underline{1 + \alpha^{10} v + \alpha^5 v^2}$	$\underline{1}$
α^6	$\alpha^1 + \alpha^{11} v + \alpha^6 v^2$	α^1
α^7	$\alpha^2 + \alpha^{12} v + \alpha^7 v^2$	α^2
α^8	$\alpha^3 + \alpha^{13} v + \alpha^8 v^2$	α^3
α^9	$\alpha^4 + \alpha^{14} v + \alpha^9 v^2$	α^4
α^{10}	$\alpha^5 + v + \alpha^{10} v^2$	α^5
α^{11}	$\alpha^6 + \alpha^1 v + \alpha^{11} v^2$	α^6
α^{12}	$\alpha^7 + \alpha^2 v + \alpha^{12} v^2$	α^7
α^{13}	$\alpha^8 + \alpha^3 v + \alpha^{13} v^2$	α^8
α^{14}	$\alpha^9 + \alpha^4 v + \alpha^{14} v^2$	α^9

The codeword $y(x)$ of the concatenated code is found in three steps. For example, assume the dataword

$$u(x) = \alpha^{14} + \alpha^3 x + \alpha^8 x^2 + \alpha^6 x^3$$

is given. In the first step, this dataword is mapped onto elements of $\mathbb{F}_{2^4}$ using the previous table:

$$u(w) = \alpha^9 + \alpha^{13} w + \alpha^3 w^2 + \alpha^1 w^3.$$

Encoding by the outer code $\mathcal{D'}_0(5,4)$ gives

$$y(w) = \alpha^9 + \alpha^{10}w + \alpha^8 w^2 + \alpha^9 w^3 + \alpha^1 w^4 \in \mathcal{D'}_0(5,4).$$

In the third step, the inverse mapping Φ^{-1} is used to map each coefficients of $y(w)$ onto its corresponding codeword (of the inner code) yielding a bivariate codeword polynomial

$$y(v,w) = \begin{array}{ccccc} \alpha^9+ & \alpha^{10}w+ & \alpha^8 w^2+ & \alpha^9 w^3+ & \alpha^1 w^4+ \\ \alpha^4 v+ & \alpha^5 vw+ & \alpha^3 vw^2+ & \alpha^4 vw^3+ & \alpha^{11} vw^4+ \\ \alpha^{14} v^2+ & 1v^2 w+ & \alpha^{13} v^2 w^2+ & \alpha^{14} v^2 w^3+ & \alpha^6 v^2 w^4 \end{array}$$

The filter bank output (codeword polynomial) $y(x)$ is obtained by transforming $v \to x$ and $w \to x^3$ (column wise).

$$\begin{aligned} y^{\Pi'}(x) = \alpha^9 + \alpha^4 x + \alpha^{14} x^2 + \alpha^{10} x^3 + \alpha^5 x^4 + x^5 + \alpha^8 x^6 + \alpha^3 x^7 + \ldots \\ \alpha^{13} x^8 + \alpha^9 x^9 + \alpha^4 x^{10} + \alpha^{14} x^{11} + \alpha^1 x^{12} + \alpha^{11} x^{13} + \alpha^6 x^{14} \end{aligned}$$

The codeword of $\mathcal{R}\,[15,10]$ is found if $vw \to x$, corresponding to a diagonal read out of the matrix defined by $y(v,w)$. In general, $v^s w^{s'}$ corresponds with x^t if $t = s \bmod 3$ and $t = s' \bmod 5$. The relation between $y^{\Pi'}(x)$ and $y(x)$ is exactly given by the interleaver $\mathbf{\Pi'}$.

$$\begin{aligned} y(x) = \alpha^9 + \alpha^5 + \alpha^{13} x^2 + \alpha^9 x^3 + \alpha^{11} x^4 + \alpha^{14} x^5 + \alpha^{10} x^6 + \alpha^3 x^7 + \ldots \\ \alpha^{14} x^8 + \alpha^1 x^9 + \alpha^4 x^{10} + 1x^{11} + \alpha^8 x^{12} + \alpha^4 x^{13} + \alpha^6 x^{14} \end{aligned}$$

The generator matrix of the concatenated code

$$\mathbf{G}_{\mathcal{D'}_0(5,4)\square\mathcal{C'}_0(3,1)} = \left[\begin{array}{ccccccccccccccc} \alpha^{10} & \alpha^5 & 0 & 0 & 0 & 1 & \alpha^{10} & 0 & 0 & 0 & \alpha^5 & 1 & 0 & 0 & 0 \\ 0 & \alpha^5 & 1 & 0 & 0 & 0 & \alpha^{10} & \alpha^5 & 0 & 0 & 0 & 1 & \alpha^{10} & 0 & 0 \\ 0 & 0 & 1 & \alpha^{10} & 0 & 0 & 0 & \alpha^5 & 1 & 0 & 0 & 0 & \alpha^{10} & \alpha^5 & 0 \\ 0 & 0 & 0 & \alpha^{10} & \alpha^5 & 0 & 0 & 0 & 1 & \alpha^{10} & 0 & 0 & 0 & \alpha^5 & 1 \end{array}\right].$$

The nonzeros of the concatenated code are $\alpha^2, \alpha^8, \alpha^{11}, \alpha^{14}$. For example, the nonzero α^8 is realized by an outer code with nonzero α^3 combined with the inner code (nonzero α^5) since $s = s' = t = 1$. All possible combinations can be found in Table 5.3. The nonzeros to obtain the RS code $\mathcal{R}\,[15,10]$ are underlined. For example, the nonzeros $\alpha^0, \alpha^9, \alpha^{12}$ are realized by band $m = 1$ ($s = 0$). The nonzeros $\alpha^{10}, \alpha^{13}, \alpha^1$ are due to band $m = 2$ ($s = 2$).

△

Table 5.3: Realization of a root of the concatenated code by combining specific roots of the inner and outer code. The nonzeros to obtain the RS code $\mathcal{R}[15, 10]$ are underlined.

s	s'	t	α^{5s}	$\alpha^{3s'}$	α^{8t}
$\underline{0}$	$\underline{0}$	$\underline{0}$	$\underline{\alpha^{5\cdot 0}}$	$\underline{\alpha^{3\cdot 0}}$	$\underline{\alpha^{0}}$
0	1	6	$\alpha^{5\cdot 0}$	$\alpha^{3\cdot 1}$	α^{3}
0	2	12	$\alpha^{5\cdot 0}$	$\alpha^{3\cdot 2}$	α^{6}
$\underline{0}$	$\underline{3}$	$\underline{3}$	$\underline{\alpha^{5\cdot 0}}$	$\underline{\alpha^{3\cdot 3}}$	$\underline{\alpha^{9}}$
$\underline{0}$	$\underline{4}$	$\underline{9}$	$\underline{\alpha^{5\cdot 0}}$	$\underline{\alpha^{3\cdot 4}}$	$\underline{\alpha^{12}}$
1	0	10	$\alpha^{5\cdot 1}$	$\alpha^{3\cdot 0}$	α^{5}
$\underline{1}$	$\underline{1}$	$\underline{1}$	$\underline{\alpha^{5\cdot 1}}$	$\underline{\alpha^{3\cdot 1}}$	$\underline{\alpha^{8}}$
$\underline{1}$	$\underline{2}$	$\underline{7}$	$\underline{\alpha^{5\cdot 1}}$	$\underline{\alpha^{3\cdot 2}}$	$\underline{\alpha^{11}}$
$\underline{1}$	$\underline{3}$	$\underline{13}$	$\underline{\alpha^{5\cdot 1}}$	$\underline{\alpha^{3\cdot 3}}$	$\underline{\alpha^{14}}$
$\underline{1}$	$\underline{4}$	$\underline{4}$	$\underline{\alpha^{5\cdot 1}}$	$\underline{\alpha^{3\cdot 4}}$	$\underline{\alpha^{2}}$
$\underline{2}$	$\underline{0}$	$\underline{5}$	$\underline{\alpha^{5\cdot 2}}$	$\underline{\alpha^{3\cdot 0}}$	$\underline{\alpha^{10}}$
$\underline{2}$	$\underline{1}$	$\underline{11}$	$\underline{\alpha^{5\cdot 2}}$	$\underline{\alpha^{3\cdot 1}}$	$\underline{\alpha^{13}}$
$\underline{2}$	$\underline{2}$	$\underline{2}$	$\underline{\alpha^{5\cdot 2}}$	$\underline{\alpha^{3\cdot 2}}$	$\underline{\alpha^{1}}$
2	3	8	$\alpha^{5\cdot 2}$	$\alpha^{3\cdot 3}$	α^{4}
2	4	14	$\alpha^{5\cdot 2}$	$\alpha^{3\cdot 4}$	α^{7}

5.4 Applications

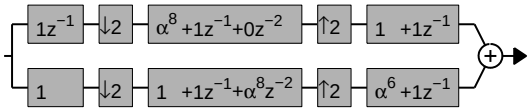
5.4.1 Recursive Filter Banks

Starting from a type II filter bank representation of an RS code, the filter bank representation can be applied recursively on the subbands, since they are (cyclic) BCH codes. An example is given in Figure 5.3, showing two filter bank representations of a rate $\frac{3}{4}$ RS code in a prime field $\mathbb{F}_{13}$. This recursive approach is important to lower the complexity of soft RS decoding.

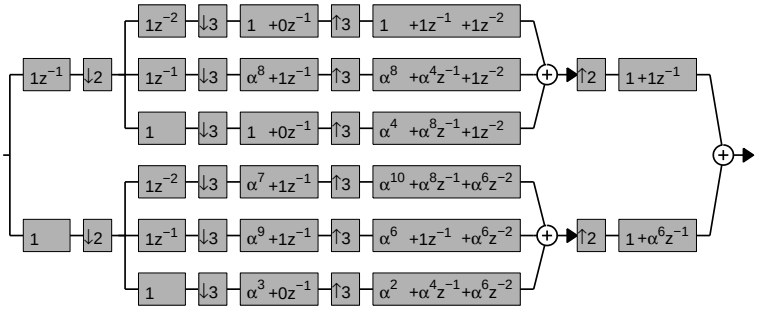
5.4.2 BCH Codes

The techniques presented in the previous section allow for an easy extension towards BCH codes. Unlike RS codes, zeros (and non-zeros) belonging to an extension field $\mathbb{F}_{p^{mn}}$ must be *grouped* such that the BCH code is defined over $\mathbb{F}_{p^m}$. Cyclotomic cosets indicate which zeros must be grouped.

More specifically, let $\mathcal{B}[\nu, \kappa]$ be a BCH code over $\mathbb{F}_{p^m}$, with $\alpha \in \mathbb{F}_{q^n}$ a primitive ν-th root of unity. Let $\nu' = \frac{\nu}{M'}$, with M' the length of the inner cyclic code. Then, M cyclotomic cosets $\mathscr{C}^{(s)}_{q,M'}$ can be found, each defining a minimal ideal (inner code) ${\mathcal{C}'}_m(M', \kappa_m) \in \mathbb{F}_q, m = 0 : M-1$. Note that, since some cyclotomic cosets might not be singletons anymore, the number of bands M differs



(a) Filter bank representation of $\mathcal{R}\,[12, 9]$ in $\mathbb{F}_{13}$.



(b) Same filter bank as in Figure 5.3(a), with each subband code further represented as a filter bank

Figure 5.3: Recursive filter bank implementation of $\mathcal{R}\,[12, 9]$.

from the length of the inner cyclic codes M'. Concerning the outer codes, they must be defined such that each outer code in combination with an inner code realizes a set of nonzeros of the concatenated code. Imagine $(\nu' + M)t \in \overline{Z}_{\mathcal{B}}$ for a certain t. Then a unique pair s and s' can be found such that

$$\begin{aligned} t &= s \bmod M' \text{ and} \\ t &= s' \bmod \nu' . \end{aligned}$$

Then, it can be similarly shown that the nonzero $\alpha^{(\nu'+M)t}$ and its conjugates are realized by concatenating an outer code with nonzeros $\mathscr{C}^{(s')}_{q^\kappa_m,\nu'}$ to the inner code with nonzeros $\mathscr{C}^{(s)}_{q,M'}$. This can be clarified in the following example.

Example 5.6 Let us construct a BCH code $\mathcal{B}\,[10, 5]$ in $\mathbb{F}_{3^2}$. Let α be a primitive 10-th root of unity, such that $\alpha = \gamma^8$, with γ a primitive 80-th root of unity in the extension field $\mathbb{F}_{3^4}$. The zeros of the code are

$$Z_{\mathcal{B}} = \{\mathscr{C}^{(0)}_{9,10}, \mathscr{C}^{(1)}_{9,10}, \mathscr{C}^{(2)}_{9,10}\} = \{8, 9, 0, 1, 2\}\,. \tag{5.14}$$

The generator polynomial $g_{\mathcal{B}}(x) = \alpha^4 + \alpha^7 x + \alpha^6 x^2 + \alpha^2 x^3 + \alpha^3 x^4 + x^5$, with roots $\alpha^{-2}, \alpha^{-1}, \alpha^0, \alpha^1, \alpha^2$. Let the length of the cyclic inner code be $M = 5$. Hence, the length of the cyclic outer code $\nu' = \frac{\nu}{M} = 2$. Since the inner code is a minimal ideal, the cyclotomic cosets over $\mathbb{F}_{3^2}$ modulo 5 are calculated:

$$\mathscr{C}^{(0)}_{9,5} = \{0\} \tag{5.15}$$

$$\mathscr{C}_{9,5}^{(1)} = \{1,4\} \tag{5.16}$$
$$\mathscr{C}_{9,5}^{(2)} = \{2,3\}\,. \tag{5.17}$$

Since $\mathscr{C}_{9,5}^{(1)}$ has 2 elements, the corresponding code $\mathcal{C}'_1$ combines 2 bands. This is also the case for $\mathscr{C}_{9,5}^{(2)}$. Therefore, the total number of bands becomes 3 instead of $M = 5$. Let us have a closer look into the second band. The corresponding inner code $\mathcal{C}'_1(5,2) \in \mathbb{F}_{3^2}$ has 3^4 codewords, and is isomorphic to $\mathbb{F}_{3^4}$. Hence, the outer code $\mathcal{D}'_1(2,1) \in \mathbb{F}_{3^4}$. The cosets to design the outer code are

$$\mathscr{C}_{81,2}^{(0)} = \{0\} \tag{5.18}$$
$$\mathscr{C}_{81,2}^{(1)} = \{1\}\,. \tag{5.19}$$

According to Eq. 5.14, to realize $\alpha^7 = \alpha^{(5+2)\cdot 1}$ $(t = 1)$ as a nonzero of the concatenated code, $\alpha^{2\cdot 1} \in \overline{Z}_{\mathcal{D}'_1}$ $(s = 1)$ and $\alpha^{5\cdot 1} \in \overline{Z}_{\mathcal{C}'_1}$ $(s' = 1)$. Hence, the nonzeros of the inner code $\mathcal{D}'_1$ are α^2 and its conjugate α^8. Consequently, with zeros $1, \alpha^4, \alpha^6$, the generator polynomial of the inner code is

$$g_{\mathcal{C}'_1}(x) = \gamma^{40} + \gamma^{10}x + \gamma^{50}x^2 + x^3 \in \mathbb{F}_{3^2}\,.$$

The isomorphism Φ between the codewords of $\mathcal{C}'_1$ and $\mathbb{F}_{3^4}$ is defined by $\Phi\gamma^{70} + \gamma^{30}x^2 + x^3 + \gamma^{40}x^4 = \gamma$. The outer code is defined by

$$g_{\mathcal{D}'_1}(x) = \gamma^{40} + x \in \mathbb{F}_{3^4}\,.$$

Similarly, the inner and outer codes in the other bands can be obtained. The nonzero α^5 is realized by the first band as a concatenation of $\mathcal{D}'_0(2,1)\square\mathcal{C}'_0(5,1)$ with

$$g_{\mathcal{C}'_0}(x) = 1 + x + x^2 + x^3 + x^4 \in \mathbb{F}_{3^2} \tag{5.20}$$
$$g_{\mathcal{D}'_0}(x) = \gamma^{40} + x \in \mathbb{F}_{3^2}\,. \tag{5.21}$$

The last band consists of a concatenation $\mathcal{D}'_2(2,1)\square\mathcal{C}'_2(5,2)$, defined as

$$g_{\mathcal{C}'_2}(x) = \gamma^{40} + \gamma^{30}x + \gamma^{70}x^2 + x^3 \in \mathbb{F}_{3^2} \tag{5.22}$$
$$g_{\mathcal{D}'_2}(x) = 1 + x \in \mathbb{F}_{3^4}\,. \tag{5.23}$$

The filter bank-like scheme is shown in Figure 5.4. Note that downsamplers are replaced by transformations in between Galois fields. However, keep in mind that a *real* filter bank structure underlies each of these bands (Section 4.4). This is illustrated in Figure 5.5 for $\mathcal{D}'_1(2,1)\square\mathcal{C}'_1(5,2)$, obtained by selecting appropriate bands from Figure 4.6 (Example 4.8).

△

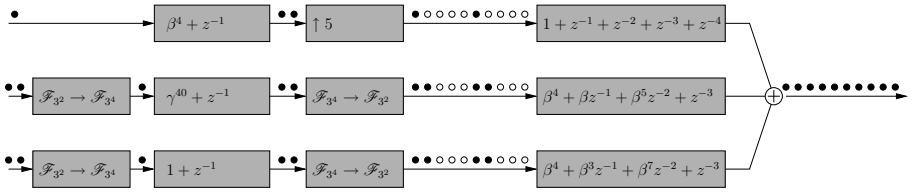


Figure 5.4: Filter bank like decomposition for $\mathcal{B}[10, 5]$ in $\mathbb{F}_{3^2}$. The subband samples are indicated with dots (zeros are represented by circles).

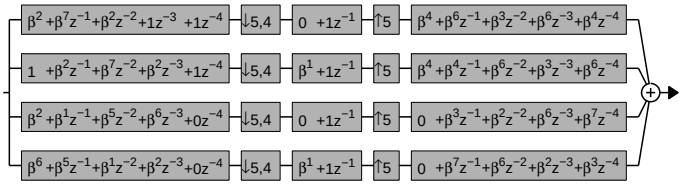


Figure 5.5: Filter bank decomposition of $\mathcal{D'}_1(2, 1)\square\mathcal{C'}_1(5, 2)$.

Example 5.7 Let us construct a BCH code $\mathcal{B}[20, 11]$ in $\mathbb{F}_3$. Let α be a primitive 20-th root of unity, such that $\alpha = \gamma^4$, with γ a primitive 80-th root of unity in the extension field $\mathbb{F}_{3^4}$. The zeros of the code are

$$Z_{\mathcal{B}} = \{\mathscr{C}^{(4)}_{3,20}, \mathscr{C}^{(11)}_{3,20}, \mathscr{C}^{(10)}_{3,20}\}. \tag{5.24}$$

The generator polynomial $g_{\mathcal{B}}(x) = i_4(x)+i_{10}(x)+i_{11}(x) = 1+x^2+x^6+x^8+x^9$, with roots $\alpha^4, \alpha^8, \alpha^{10}, \alpha^{11}, \alpha^{12}, \alpha^{13}, \alpha^{16}, \alpha^{17}, \alpha^{19}$. Let the length of the cyclic inner code be $M = 4$. Hence, the length of the cyclic outer code $\nu' = \frac{\nu}{M} = 5$. Since the inner code is a minimal ideal, the cyclotomic cosets over $\mathbb{F}_3$ modulo 4 are calculated:

$$\mathscr{C}^{(0)}_{3,4} = \{0\} \tag{5.25}$$

$$\mathscr{C}^{(1)}_{3,4} = \{1, 3\} \tag{5.26}$$

$$\mathscr{C}^{(2)}_{3,4} = \{2\}. \tag{5.27}$$

Since $\mathscr{C}^{(1)}_{3,4}$ has 2 elements, the corresponding code $\mathcal{C'}_1$ combines 2 bands. Therefore, the total number of bands becomes 3 instead of $M = 4$. Another consequence of this coset with multiple elements is that the corresponding inner code $\mathcal{C'}_1(4, 2) \in \mathbb{F}_3$ has 3^2 codewords, and is isomorphic to $\mathbb{F}_{3^2}$. Hence, the outer code $\mathcal{D'}_1 \in \mathbb{F}_{3^2}$. The cosets to design the outer code are

$$\mathscr{C}^{(0)}_{9,5} = \{0\} \tag{5.28}$$

$$\mathscr{C}^{(1)}_{9,5} = \{1, 4\} \tag{5.29}$$

Table 5.4: Realization of a root of the concatenated code by combining specific roots of the inner and outer code. The nonzeros to obtain the BCH code $\mathcal{B}[10,5]$ of Example 5.6 are underlined.

s	s'	t	α^{2s}	$\alpha^{5s'}$	α^{7t}
0	0	0	$\alpha^{2\cdot 0}(\mathscr{C}_{9,5}^{(0)})$	$\alpha^{5\cdot 0}(\mathscr{C}_{9,2}^{(0)})$	$\alpha^{0}(\mathscr{C}_{9,10}^{(0)})$
$\underline{0}$	$\underline{1}$	$\underline{5}$	$\underline{\alpha^{2\cdot 0}(\mathscr{C}_{9,5}^{(0)})}$	$\underline{\alpha^{5\cdot 1}(\mathscr{C}_{9,2}^{(1)})}$	$\underline{\alpha^{5}(\mathscr{C}_{9,10}^{(5)})}$
1	0	6	$\alpha^{2\cdot 1}, \alpha^{2\cdot 4}(\mathscr{C}_{9,5}^{(1)})$	$\alpha^{5\cdot 0}(\mathscr{C}_{81,2}^{(0)})$	$\alpha^{2}\alpha^{8}(\mathscr{C}_{9,10}^{(2)})$
$\underline{1}$	$\underline{1}$	$\underline{1}$	$\underline{\alpha^{2\cdot 1}, \alpha^{2\cdot 4}(\mathscr{C}_{9,5}^{(1)})}$	$\underline{\alpha^{5\cdot 1}(\mathscr{C}_{81,2}^{(1)})}$	$\underline{\alpha^{3}, \alpha^{7}(\mathscr{C}_{9,10}^{(3)})}$
$\underline{2}$	$\underline{0}$	$\underline{2}$	$\underline{\alpha^{2\cdot 2}, \alpha^{2\cdot 3}(\mathscr{C}_{9,5}^{(2)})}$	$\underline{\alpha^{5\cdot 0}(\mathscr{C}_{81,2}^{(0)})}$	$\underline{\alpha^{4}, \alpha^{6}(\mathscr{C}_{9,10}^{(4)})}$
2	1	7	$\alpha^{2\cdot 2}, \alpha^{2\cdot 3}(\mathscr{C}_{9,5}^{(2)})$	$\alpha^{5\cdot 1}(\mathscr{C}_{81,2}^{(1)})$	$\alpha^{1}, \alpha^{9}(\mathscr{C}_{9,10}^{(1)})$

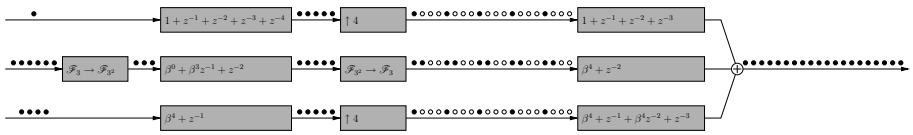


Figure 5.6: Filter bank like decomposition for $\mathcal{B}[20,11]$ in $\mathbb{F}_3$. The subband samples are indicated with dots (zeros are represented by circles).

$$\mathscr{C}_{9,5}^{(2)} \quad = \quad \{2,3\}\,. \tag{5.30}$$

According to Eq. 5.14, to realize α as a nonzero of the concatenated code, $1 \in \overline{Z}_{\mathcal{D}'_1}$ and $1 \in \overline{Z}_{\mathcal{C}'_1}$. To realize α^5, 0 should be also a member of $\overline{Z}_{\mathcal{D}'_1}$. Consequently,

$$g_{\mathcal{C}'_1}(x) \quad = \quad \gamma^{40} + x^2 \in \mathbb{F}_3 \tag{5.31}$$

$$g_{\mathcal{D}'_1}(x) \quad = \quad 1 + \gamma^{30}x + x^2 \in \mathbb{F}_{3^2}\,. \tag{5.32}$$

The isomorphism between the codewords of $\mathcal{C}'_1$ is given in Table 5.5. Similarly, the rest of the outer codes can be obtained. The nonzeros α^0 and α^2 can be realized by a concatenation of $\mathcal{D}'_0(5,1)\square\mathcal{C}'_0(4,1)$ and $\mathcal{D}'_2(5,4)\square\mathcal{C}'_2(4,1)$, respectively. The filter bank scheme is shown in Figure 5.6.

△

5.4.3 Unequal Error Protecting Codes

In a number of important applications, not all data are of equal value to the user; e.g. a mistake in the IP-address of a packet renders the packet completely useless. Therefore, unequal error protecting codes are of interest, that protect some symbols against a greater number of errors than others. Masnick

Table 5.5: Isomorphism Φ between codewords of the inner code, which is a minimal ideal, and $\mathbb{F}_{3^2}$. The idempotent $i_1(v)$ is underlined.

dataword	codeword $y_{\mathcal{C}'_1}(v)$	$\Phi y_{\mathcal{C}'_1}(v) \in \mathbb{F}_{3^2}$
0	0	0
v	$\gamma^{40}v + v^3$	γ^{20}
$\gamma^{40}v$	$v + \gamma^{40}v^3$	γ^{60}
$\underline{1}$	$\underline{\gamma^{40} + v^2}$	$\underline{1}$
$1 + v$	$\gamma^{40} + \gamma^{40}v + v^2 + v^3$	γ^{70}
$1 + \gamma^{40}v$	$\gamma^{40} + v + v^2 + \gamma^{40}v^3$	γ^{50}
γ^{40}	$1 + \gamma^{40}v^2$	γ^{40}
$\gamma^{40} + v$	$1 + \gamma^{40}v + \gamma^{40}v^2 + v^3$	γ^{10}
$\gamma^{40} + \gamma^{40}v$	$1 + v + \gamma^{40}v^2 + \gamma^{40}v^3$	γ^{30}

Table 5.6: Realization of a root of the concatenated code by combining specific roots of the inner and outer code. Again, the nonzeros of the BCH code $\mathcal{B}[20, 11]$ of Example 5.7 are underlined.

s	s'	t	α^{5s}	$\alpha^{4s'}$	α^{9t}
0	0	0	$\alpha^{5\cdot 0}(\mathscr{C}^{(0)}_{3,4})$	$\alpha^{4\cdot 0}(\mathscr{C}^{(0)}_{3,5})$	$\alpha^{0}(\mathscr{C}^{(0)}_{3,20})$
$\underline{0}$	$\underline{1}$	$\underline{4}$	$\underline{\alpha^{5\cdot 0}(\mathscr{C}^{(0)}_{3,4})}$	$\underline{\alpha^{4\cdot 1}, \alpha^{4\cdot 2}, \alpha^{4\cdot 3}, \alpha^{4\cdot 4}(\mathscr{C}^{(1)}_{3,5})}$	$\underline{\alpha^{4}, \alpha^{8}, \alpha^{12}, \alpha^{16}(\mathscr{C}^{(4)}_{3,20})}$
1	0	5	$\alpha^{5\cdot 1}, \alpha^{5\cdot 3}(\mathscr{C}^{(1)}_{3,4})$	$\alpha^{4\cdot 0}(\mathscr{C}^{(0)}_{9,5})$	$\alpha^{5}\alpha^{15}(\mathscr{C}^{(5)}_{3,20})$
1	1	1	$\alpha^{5\cdot 1}, \alpha^{5\cdot 3}(\mathscr{C}^{(1)}_{3,4})$	$\alpha^{4\cdot 1}, \alpha^{4\cdot 3}(\mathscr{C}^{(1)}_{9,5})$	$\alpha^{1}, \alpha^{3}, \alpha^{7}, \alpha^{9}(\mathscr{C}^{(1)}_{3,20})$
$\underline{1}$	$\underline{2}$	$\underline{11}$	$\underline{\alpha^{5\cdot 1}, \alpha^{5\cdot 3}(\mathscr{C}^{(1)}_{3,4})}$	$\underline{\alpha^{4\cdot 2}(\mathscr{C}^{(2)}_{9,5})}$	$\underline{\alpha^{11}, \alpha^{13}, \alpha^{17}, \alpha^{19}(\mathscr{C}^{(11)}_{3,20})}$
$\underline{2}$	$\underline{0}$	$\underline{10}$	$\underline{\alpha^{5\cdot 2}(\mathscr{C}^{(2)}_{3,4})}$	$\underline{\alpha^{4\cdot 0}(\mathscr{C}^{(0)}_{3,5})}$	$\underline{\alpha^{10}(\mathscr{C}^{(10)}_{3,20})}$
2	1	2	$\alpha^{5\cdot 2}(\mathscr{C}^{(2)}_{3,4})$	$\alpha^{4\cdot 1}, \alpha^{4\cdot 2}, \alpha^{4\cdot 3}, \alpha^{4\cdot 4}(\mathscr{C}^{(1)}_{3,5})$	$\alpha^{2}, \alpha^{6}, \alpha^{14}, \alpha^{18}(\mathscr{C}^{(2)}_{3,20})$

and Wolf [48] were the first to introduce the concept of unequal error protection. To formally define this concept, we follow the definitions of Dunning and Robbins [21]. They introduced the concept of a separation vector:

Definition 5.6 (Separation vector) For a linear code $\mathcal{L}[\nu, \kappa]$ with generator matrix $\mathbf{G}_{\mathcal{L}}$, the $1 \times \kappa$ separation vector $\mathbf{s}$ w.r.t. this generator matrix is defined by

$$s[i] = \min(\mathrm{w}_H(\mathbf{G}_{\mathcal{L}}\mathbf{u}) | \mathbf{u} \in \mathbb{F}_q, u[i] \neq 0).$$

It is straightforward that the minimum Hamming distance of the code $\delta = \min(\mathbf{s})$. □

This means that the i-th symbol can always be corrected as long as the error pattern has a (Hamming) weight less than or equal to $\lfloor \frac{s[i]-1}{2} \rfloor$ (assume maximum likelihood nearest neighbor decoding). So, with the errors in certain

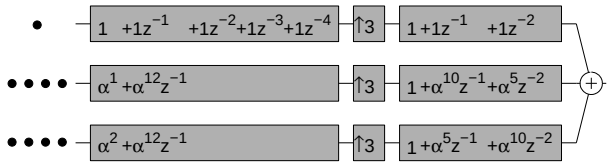


Figure 5.7: Critically subsampled ($M = 3$) filter bank for an unequal error protecting $\mathcal{C}(15, 9)$ code, found in Example 5.8

positions, it is possible to correct more errors than predicted by the minimum Hamming distance δ [10]. The separation vector depends upon the generator matrix. Fortunately, every codes has a so-called optimal generator matrix [74, 21], whose separation vector is component-wise larger than or equal to the separation vector of any other generator matrix of the code. Therefore, the separation vector *of a linear code* is defined as the separation vector of an optimal generator matrix of this code. To specify the separation vector of a linear code, the notation $\mathcal{L}[\nu, \kappa, \mathbf{s}]$ is used. An important problem is the construction of such unequal error protecting codes. It was noted by Van Gils [75] that concatenated codes in the sense of Forney offer a possible solution. The next example shows that filter banks can be used to construct such unequal error protecting codes. In this application, some subband (outer) codes have a distance which is larger than others, thereby offering stronger protection to the corresponding part of the dataword. The roots of the code are deliberately unequally distributed over the bands, thereby destroying the RS or BCH code structure.

Example 5.8 In this example, a code $\mathcal{C}(15, 9)$ will be constructed in $\mathbb{F}_{2^4}$. It is a cyclic code with roots $\{\alpha^0, \alpha^3, \alpha^4, \alpha^5, \alpha^6, \alpha^9\}$. The roots are not subsequent powers of α, such that this is *not* an RS code. In fact there exists a codeword of weight 5, while the corresponding RS code $\mathcal{C}(15, 9)$ has a Hamming distance of 7. A critically subsampled filter bank for $\mathcal{C}(15, 9)$ as shown in Figure 5.7. The roots $\alpha^0, \alpha^3, \alpha^6, \alpha^9$ are all allocated to the first subband filter, making it a strong repetition code. The roots α^4 and α^5 are roots of the subband filters in the second and third band. As can be expected, this code protects $\mathbf{u}(0)$ stronger against errors than the other data symbols. For this example, this is checked by calculating the separation vector

$$\mathbf{s} = \begin{bmatrix} 6 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 \end{bmatrix}. \tag{5.33}$$

This means that if the first data symbol is changed, the codeword will change in at least 6 positions. Changing any other dataword symbol however leads to a change in only 5 positions.

△

5.5 Conclusion

In this chapter, the relation between the filter bank representation and the 2-D cyclic structure of RS and BCH codes is discussed. The starting point is a slightly modified filter bank, in which each component code (subband code or synthesis bank code) is cyclic. Each band can be seen as a concatenation (in the sense of Forney) of these two codes, and is shown to be cyclic itself due to the isomorphism between the codewords of the inner code and the elements of $\mathbb{F}_{q^\kappa}$. Moreover, the main result shows a well defined relationship between a nonzero of the RS code and a nonzero of the subband code and the synthesis bank code.

These additional insights lead to a number of interesting conclusions. It is seen how the filter bank theory can be applied recursively on the subband codes (which are seen to be cyclic). Secondly, it is illustrated how the theory of 2-D cyclic codes extends towards BCH codes: instead of working with single nonzeros of the RS code, certain nonzeros should be grouped into sets (cyclotomic cosets). Finally, we have a look at unequal error protecting codes.

Part II

Applications

Chapter 6

Soft-In Soft-Out Reed-Solomon Decoding

6.1 Introduction

Unlike in image and audio applications, a filter describing an RS code is not very long and therefore, *encoding* a dataword by using a filter bank representation is not justifiable from a complexity point of view. The problem however resides at the *decoder*, which exploits the redundancy to correct errors in a possibly corrupted codeword. The importance of an efficient RS decoder nowadays can hardly be underestimated. Their applications span from storage devices (including magnetic tape, CD, DVD, barcodes, etc) over high-speed modems (such as ADSL, VDSL, etc), wireless and mobile communications (including cellular telephones, microwave links, etc) to satellite and deep space communications. Therefore, in the literature, a vast number of research results on decoding RS codes is found: algebraic decoding algorithms are readily available for hard-decision decoding, e.g. Berlekamp-Massey's algorithm and Guruswami-Sudan's list decoding algorithm [37]. The potential benefit of soft decision RS decoding is roughly 2 dB, as discussed in [11]. 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 codes (and other powerful block codes) is however much more complex than hard decision decoding. As predicted by the Cut-Set bound [81, 28], the exact MAP-solution can only be found at a prohibitively large complexity. The search for low-complexity soft RS decoding algorithms has attracted a lot of attention recently, as can be illustrated by a short overview of algorithms published in the literature. The trellis based schemes of Hage-

nauer [38] 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 [31], based on ordered statistics. An alternative approach to the problem of efficient soft decoding is the generalized minimum distance decoding (GMD) of Forney [29]. 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 [44], resulting in a *Soft In Hard Out* (SIHO) algorithm. However, Benedetto, Divsalar and Hagenauer describe in [4] 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 equalization, demodulation and channel decoding. Hence, we state that an efficient SISO RS decoder is desirable.

Recently, this problem of efficient SISO decoding was solved for concatenated codes. The subcodes are then decoded one after the other, according to Gallager's algorithm, thereby reducing the complexity. Concatenated coding was first introduced for block codes by Forney [27] 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 [3] resulting in Turbo codes [7]. Recent work by Hagenauer [38] 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 concatenated code, e.g. the minimum distance between the codewords.

In this chapter, however, the opposite approach is taken. Starting from an overall RS code, the previously discussed filter bank representation is applied to decompose the code into many smaller subcodes. Gallager's algorithm can then be applied to obtain a SISO decoder for the overall code. With the turbo principle in mind, several other RS code decompositions have appeared in the literature. In [77], Vardy presents a decomposition of an RS code in the binary subfield. This decomposition in the binary field involves a number of BCH codes. However, these BCH codes are not able to span the complete RS code space exactly and 'glue' vectors are needed to glue the subcodes together. This decomposition lead to a number of SIHO RS decoding algorithms as described by Vardy and Be'ery [77] and Ponnampalam and Vucetic [55]. With optimal SISO decoders for each BCH code and the glue code, a SISO RS decoder can be built based on Vardy's decomposition, see e.g. Ponnampalam and Grant [54] and Halford *et al.* [39]. An important contribution of the last paper is that the glue code itself has a trellis structure, just as the BCH codes, and can consequently be optimally SISO decoded with BCJR's algorithm. Since the corresponding Tanner graph is cycle free, the exact MAP solution is found. However, the resulting algorithm will have a prohibitively large complexity, as stated by the Cut-Set bound [81, 28]. Therefore, there is a lot of interest into suboptimal SISO RS decoding algorithms, for example by tolerating a number

of cycles in the Tanner graph. Most of these SISO algorithms are working with the binary image of RS codes, e.g. the ordered statistics approach of Fossorier and Lin [31]. This method sorts the received bits with respect to their reliability. The columns of the generator matrix corresponding to the most reliable bits are then reduced to an identity matrix. This is comparable to the adaptive belief propagation algorithm presented by Jiang and Narayanan [42]. In this algorithm, Gallager's algorithm is run on an adapted parity check matrix, where the columns of the least reliable bits are reduced to an identity matrix. In [22], this approach is also combined with algebraic soft-decision decoding [44].

Unlike these algorithms, all working with a *binary image* of an RS code, our filter bank representation offers a decomposition in the original *Galois field*. This is important, since a binary image is often used to circumvent soft-decoding (see Section 1.1.3). Moreover, if higher order modulations are used, this conversion to soft-bits incurs information loss. The decoder presented here fully operates in the original GF. The subband codes can be easily SISO decoded and are operated together in a turbo-like fashion to build a SISO decoder for the overall code. This method, unlike previously mentioned methods that appeared in literature, results in a decomposition with a limited amount of cycles in the corresponding Tanner graph. These cycles are comparable with the cycles in a turbo code, resulting in a suboptimal RS decoder but with a feasible complexity. The goal of this chapter is to illustrate the decomposition, and to serve as a proof of concept of how this decomposition can be used to build a SISO decoder.

This chapter is structured as follows: Following an introduction on optimal decoding criteria (Section 6.2), the MAP criterium is selected and Gallager's algorithm is presented as a suboptimal decoding technique in Section 6.3. In Section 6.4, it is seen how Gallager's algorithm is deployed to make a soft RS decoder using the filter bank decomposition. Finally, some directions for further research are discussed.

6.2 Optimal Decoding Criteria

Before looking into suboptimal decoding strategies, this section elaborates more on optimal (but infeasible from a complexity point of view) decoders. Note that the techniques presented in this section are generally applicable to any (possibly nonlinear) block code $\mathcal{C}(\nu, M, \delta)$. Consider the situation in Figure 6.1. The message source selects a message (out of M possible messages) which is mapped onto its corresponding codeword $\mathbf{y}$. The discrete memoryless channel adds uncertainty, and the vector $\tilde{\mathbf{y}}$ is observed. Assume the channel is completely characterized by its transition probabilities $\Pr(\tilde{y}[i] \mid y[i])$.

The decoder aims to find $\hat{\mathbf{y}} \in \mathcal{C}(\nu, M, \delta)$. The probability of making a correct

decision is equal to

$$\Pr(\mathbf{y} = \hat{\mathbf{y}}) = \Pr(\mathbf{y}|\tilde{\mathbf{y}}) \Pr(\tilde{\mathbf{y}}) .$$

The optimum decision device observes the received vector $\tilde{\mathbf{y}}$, and chooses the codeword $\mathbf{y}$ to maximize the probability of a correct decision:

$$\hat{\mathbf{y}} = \arg \max_{\mathbf{y} \in \mathcal{C}(\nu, M, \delta)} \Pr(\mathbf{y}|\tilde{\mathbf{y}}) \Pr(\tilde{\mathbf{y}}) . \tag{6.1}$$

If ties occur in this assignment, an arbitrary choice among those decisions that are tied can be made. Since the second factor $\Pr(\tilde{\mathbf{y}})$ in Eq. 6.1 is independent of $\mathbf{y}$, the objective function can be simplified to $\Pr(\mathbf{y}|\tilde{\mathbf{y}})$. The latter is called the *a posterior probability*. Hence, the optimum detector is referred to as the *Maximum A Posteriori* (MAP) detector.

Definition 6.1 (MAP Detector) The Maximum A Posteriori' detector is defined as the detector that chooses the codeword $\mathbf{y}$ to maximize the a posteriori' probability $\Pr(\mathbf{y}|\tilde{\mathbf{y}})$ given a received vector $\tilde{\mathbf{y}}$. □

Theorem 6.2 *(MAP detection rule)*

$$\hat{\mathbf{y}} = \arg \max_{\mathbf{y} \in \mathcal{C}(\nu, M, \delta)} Pr(\mathbf{y}|\tilde{\mathbf{y}}) . \tag{6.2}$$

The a posteriori' probability can be easily found using Bayes Rule

$$\Pr(\mathbf{y}|\tilde{\mathbf{y}}) = \frac{\Pr(\tilde{\mathbf{y}}|\mathbf{y}) \Pr(\mathbf{y})}{\Pr(\tilde{\mathbf{y}})}$$

and the channel transition probabilities:

$$\Pr(\tilde{\mathbf{y}}|\mathbf{y}) = \prod_{i=0}^{\nu-1} \Pr(\tilde{y}[i] \mid y[i]) .$$

If the a priori probabilities are equal ($\Pr(\mathbf{y}) = \frac{1}{M}$), as is typically assumed in digital transmission (otherwise, the message should be further coded), the optimal policy is to maximize $\Pr(\tilde{\mathbf{y}}|\mathbf{y})$. In this case, the MAP detector becomes the Maximum Likelihood (ML) Detection Rule:

Theorem 6.3 *(ML detection rule)*

$$\hat{\mathbf{y}} = \arg \max_{\mathbf{y} \in \mathcal{C}(\nu, M, \delta)} Pr(\tilde{\mathbf{y}}|\mathbf{y}) . \tag{6.3}$$

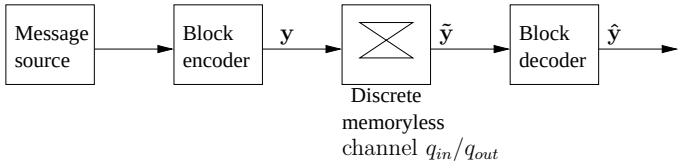


Figure 6.1: Block coding over the discrete memoryless channel.

Again, in the case of ties, an arbitrary tie breaking rule is acceptable. As with the MAP detector, the Maximum Likelihood (ML) decoder hence minimizes the probability of error $\Pr_w$ by choosing a codeword $\hat{\mathbf{y}} \in \mathcal{C}(\nu, M, \delta)$ such that $\Pr(\tilde{\mathbf{y}}|\mathbf{y})$ is maximized.

But the codeword now only depends on the channel transition probabilities and is independent of the input distribution (by assumption). The ML detector essentially ignores the $\frac{1}{M}$ factor in Eq. 6.2. This type of detector only minimizes $\Pr_w$ when the input data symbols have equal probability of occurrence. As this requirement is often met in practice, ML detection is often used. Even when the input distribution is not uniform, ML detection is still often employed as a detection rule, because the input distribution may be unknown and thus assumed to be uniform.

Let us first consider an example of ML decoding on the BSC, with crossover probability ϵ. Therefore, we need the following definition

Definition 6.4 The Hamming weight of a vector $\mathrm{w}_H(\mathbf{x})$ is the number of nonzero $x(i)$. The Hamming distance between two vectors $\mathrm{w}_H(\mathbf{x}, \mathbf{y}) = \mathrm{w}_H(\mathbf{x} - \mathbf{y})$, i.e. the number of positions where $\mathbf{x}$ and $\mathbf{y}$ differ. □

With this definition, $\Pr(\tilde{\mathbf{y}}|\mathbf{y})$ is given by

$$\Pr(\tilde{\mathbf{y}}|\mathbf{y}) = \prod_{i=0}^{\nu-1} \Pr(\tilde{y}[i] \mid y[i]) = \epsilon^{d_H(\tilde{\mathbf{y}},\mathbf{y})}(1-\epsilon)^{\nu - d_H(\tilde{\mathbf{y}},\mathbf{y})}.$$

A brute force decoding scheme, called *nearest neighbor decoding*, simply compares the received word $\tilde{\mathbf{y}}$ with each codeword and picks the codeword that is closest, in terms of Hamming distance. Hamming distance is a true distance in mathematical sense, i.e. it is nonnegative, symmetric and fulfills the triangle inequality. It is seen that this metric follows naturally from the application of ML-decoding to the BSC. Hence, it is an important code design criterium and explains the success of RS codes (which maximize the Hamming distance). It can be shown that *nearest neighbor decoding* using the Hamming distance is optimal in terms of minimizing the word error rate $\Pr_w$ for *any* q-ary sym-

metric channel [67]. For non-symmetric channels however, the metric is more complex. In summary,

Theorem 6.5 *(q-SC ML detection rule)*

$$\hat{\mathbf{y}} = \arg \min_{\mathbf{y} \in \mathcal{C}(\nu, M, \delta)} d_H(\tilde{\mathbf{y}}, \mathbf{y}) \,. \tag{6.4}$$

So far, the physical origin of the channels is ignored. Since most transmission media and their noise are continuous in nature, the discrete nature of these channels seems strange at first sight. In reality however, these channels arise from a concatenation of a continuous transmission medium with a hard decision demodulator, e.g. an impulse noise channel followed by a hard decision device – as frequently encountered in wired communications – can be approximated with a q-ary symmetric channel. In this case, a codeword symbol is not corrupted with noise with a high probability $1 - \epsilon$. Conversely, if the symbol is corrupted with noise, it is completely lost and a random symbol is received that has no connection whatsoever with the transmitted symbol. I.e., the received symbol does not carry information about the transmitted symbol.

Likewise, an AWGN channel with antipodal modulation and a $\operatorname{sgn}(\bullet)$ function in the demodulator behaves as a BSC. However, in the AWGN case, such a hard decision demodulator tends to throw away quite some (useful) information. Indeed, even if the received symbol is wrong, the correct symbol is found with a high probability in the neighborhood of the received symbol. A decoder that exploits this information by using finely quantized demodulator outputs is known as a *soft-decision decoder*. I.e. a soft demodulator with a high resolution $\frac{1}{\Delta y}$ is used, producing additional reliability information. Let us give an example to clearly distinguish between hard and soft decoding.

Example 6.1 Assume that the code defined in Example 2.3 is used to transmit uniformly distributed data symbols over an AWGN channel with $\sigma = 0.7$. Imagine the corrupted codeword

$$\tilde{\mathbf{y}} = \begin{bmatrix} -1.3028 & -0.1659 & 0.0877 & -0.7986 \end{bmatrix}$$

is received, corresponding to the codeword

$$\mathbf{y} = \begin{bmatrix} -1 & 1 & 0 & -1 \end{bmatrix}$$

being sent. Three scenarios are considered:

- In the first scenario (See also Figure 6.2), a hard decision device is added at the receiver rounding the received codeword:

$$\tilde{\mathbf{y}} = \begin{bmatrix} -1 & 0 & 0 & -1 \end{bmatrix} .$$

The super channel, consisting of the AWGN channel and the decision device, is a non-symmetric ternary discrete memoryless channel. Approximating this channel with a symmetric one, the transition probabilities $\Pr(\tilde{y}[i] \mid y[i])$ are shown in Table 6.1.

Table 6.1: Channel transition probabilities $\Pr(\tilde{y}[i] \mid y[i])$ for the first scenario

$\tilde{y}[i] \backslash y[i]$	-1	0	1
-1	0.7	0.15	0.15
0	0.15	0.7	0.15
1	0.15	0.15	0.7

With this table, the probabilities $\Pr(\tilde{\mathbf{y}}|\mathbf{y})$ are easily calculated (See Table 6.2). Applying the ML detection rule (Eq. 6.3) results in

$$\hat{\mathbf{y}} = \begin{bmatrix} -1 & 1 & 0 & -1 \end{bmatrix},$$

which is the correct codeword. It is important to realize that the thus obtained codeword $\hat{\mathbf{y}}$ does *not* depend on the crossover probability ϵ as long as it is small enough ($2\epsilon < 1 - \epsilon$). Due to the symmetry in the channel, ML-decoding corresponds with nearest neighbor decoding in terms of Hamming distance, which is indeed independent of ϵ.

Table 6.2: Codeword metrics for the code of Example 2.3 in the three scenarios.

y_0	y_1	y_2	y_3	scenario 1	scenario 2	scenario 3
0	0	0	0	0.0110	0.0155	0.2084
0	1	-1	1	0.0005	0.0002	0.0011
0	-1	1	-1	0.0024	0.0089	0.0836
1	1	1	0	0.0005	0.0002	0.0006
1	-1	0	1	0.0024	0.0000	0.0002
1	0	-1	-1	0.0110	0.0014	0.0029
-1	-1	-1	0	0.0024	0.0089	0.1635
-1	0	1	1	0.0110	0.0014	0.0327
-1	**1**	**0**	**-1**	**0.0515**	**0.0676**	**0.5070**

- The non-symmetric (exact) channel probabilities for the aforementioned super channel can be easily obtained from the Gaussian cumulative distribution function $\mathcal{CN}$, e.g.

$$\begin{aligned} \Pr(\tilde{y}[i] = -1 \mid y[i] = -1) &= \mathcal{CN}(-0.5 + 1) - \mathcal{CN}(-\infty + 1) = 0.7625 \\ \Pr(\tilde{y}[i] = -1 \mid y[i] = 0) &= \mathcal{CN}(0.5 + 1) - \mathcal{CN}(-0.5 + 1) = 0.2215 \end{aligned}$$

$$\Pr\left(\tilde{y}\left[i\right]=-1|\, y\left[i\right]=1\right) \quad = \quad \mathcal{CN}(\infty+1)-\mathcal{CN}(0.5+1)=0.0161.$$

The updated metrics can be found in Table 6.2, under scenario 2. Again, applying the ML detection rule (Eq. 6.3) results in a correct decision.

Table 6.3: Channel transition probabilities $\Pr\left(\tilde{y}\left[i\right]\,|\,y\left[i\right]\right)$ for the second scenario

$\tilde{y}\left[i\right]\backslash y\left[i\right]$	-1	0	1
-1	0.7625	0.2375	0.0161
0	0.2215	0.5249	0.2215
1	0.0161	0.2375	0.7625

The resolution of the decision device plays a crucial role in the performance of the decoder, as seen in the next scenario. It is important to note that - since this resolution $\Delta y=1$ is not larger than the transmitter resolution - the decoder described above is still viewed as a hard decoder, regardless of the fact that there does not exist an easy distance measure corresponding to the metric.

- If the resolution of the decision device is increased,

$$\begin{aligned}\Pr\left(\tilde{y}\left[i\right]\,|\,y\left[i\right]\right) &= \mathcal{CN}(y\left[i\right]-\tilde{y}\left[i\right]+\frac{\Delta y}{2})-\mathcal{CN}(y\left[i\right]-\tilde{y}\left[i\right]-\frac{\Delta y}{2})\\ &\approx \mathcal{N}(y\left[i\right]-\tilde{y}\left[i\right])\Delta y\end{aligned}$$

in which the Gaussian probability density function

$$\mathcal{N}(y\left[i\right]-\tilde{y}\left[i\right])=\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{(y[i]-\tilde{y}[i])^2}{2\sigma^2}}$$

In the limit with $\Delta y\rightarrow 0$ (infinite resolution), the decoder uses all the information available at the receiver and is called a *soft decoder*. The ML solution requires the calculation of

$$\Pr\left(\tilde{\mathbf{y}}|\mathbf{y}\right) = \prod_{i=0}^{\nu-1}\Pr\left(\tilde{y}\left[i\right]\,|\,y\left[i\right]\right) \tag{6.5}$$

$$= \frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{||\mathbf{y}-\tilde{\mathbf{y}}||^2}{2\sigma^2}}, \tag{6.6}$$

for each codeword $\mathbf{y}\in\mathcal{C}\left(4,9,3\right)$ (last column in Table 6.2). As can be concluded from this table, the correct codeword is found. More importantly, Eq. 6.5 shows that the distance corresponding to this metric is the Euclidean distance. In other words, a soft ML decoder aims to find

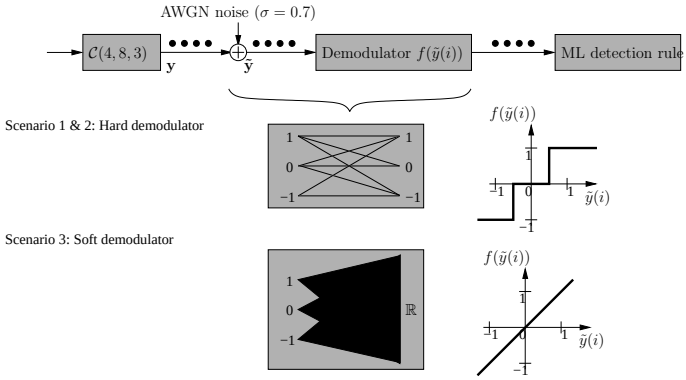


Figure 6.2: Three decoding scenarios considered in Example 2.3.

the codeword $\mathbf{y}$ that is closest (in terms of Euclidean distance) to the received word $\tilde{\mathbf{y}}$, at least in the AWGN case. In Figure 6.3, simulation results showing the performance of the decoders in each of the three scenarios can be found.

△

For a q-ary symmetric channel with hard decoding, the decision rule corresponds with the Hamming distance. If the channel is an AWGN channel with soft decoding, the decision rule can be expressed in terms of Euclidean distance.

Theorem 6.6 *(AWGN MAP detection rule)*

$$\hat{\mathbf{y}} = \arg \max_{\mathbf{y} \in \mathcal{C}(\nu, M, \delta)} ||\mathbf{y} - \tilde{\mathbf{y}}||^2 - 2\sigma^2 \log_e(Pr(\mathbf{y})) \,. \tag{6.7}$$

Similarly, if the codewords are equally likely,

Theorem 6.7 *(AWGN ML detection rule)*

$$\hat{\mathbf{y}} = \arg \max_{\mathbf{y} \in \mathcal{C}(\nu, M, \delta)} ||\mathbf{y} - \tilde{\mathbf{y}}||^2 \,. \tag{6.8}$$

6.3 MAP Estimate and Gallager's Algorithm

A SISO decoder is aiming for the MAP estimate of a variable $\mathbf{y}$. In the rest of this chapter, the MAP estimate on a *symbol level* is aimed for, which

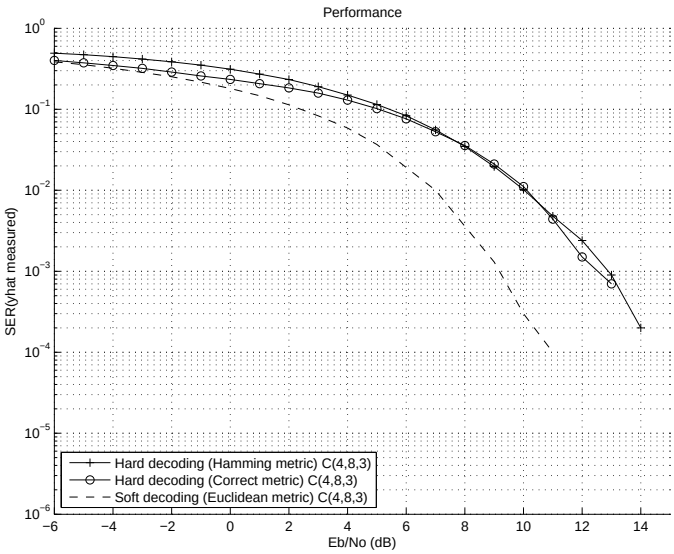


Figure 6.3: Simulation results for the code of Example 2.3 in the three scenarios as discussed in Example 6.1.

is a marginalized version of the one presented in Definition 6.1. The exact (symbol-wise) MAP estimate can be obtained by a procedure often referred to as *exhaustive combination and marginalization* [39]. This procedure, called $SISO_{\text{Exhaustive}}$, has 2 inputs: $\mathbf{P}^{y,Prio}$ essentially is a $\nu \times q$ probability matrix with the i,j-th entry indicating the probability that $y[i]$ equals the j-th element of $\mathbb{F}_q$ given the channel output (noisy observation) $\tilde{y}[i]$.

$$\mathbf{P}^{y,Prio}[i,j] = \Pr\left(y[i] = F^{-1}(j) \mid \tilde{y}[i]\right), \forall i = 0..\nu-1, j = 0..q-1 \qquad (6.9)$$

In this equation, the mapping F is defined by

$$F : \mathbb{F}_q \to \mathbb{N} : \left\{ \begin{array}{lcl} 0 & \to & 0 \\ \alpha^j & \to & j+1, j = 0..q-2 \end{array} \right.$$

The second argument is an $(\nu-\kappa) \times \nu$ parity check matrix $\mathbf{H}_{\mathcal{L}}$ of a linear code $\mathcal{L}[\nu,\kappa]$. The procedure $SISO_{\text{Exhaustive}}$, which is also the main building block in a message passing decoder, consists of two steps (algorithm 1). In step 1, $\Pr(\mathbf{y}|\mathbf{H}_{\mathcal{L}}), \forall \mathbf{y}$ is calculated. Next, the $\nu \times q$ reliability matrix holding marginal a posteriori' probabilities is calculated:

$$\mathbf{P}^{y,Post}[i,j] = \Pr\left(y[j] = F^{-1}(f)|\mathbf{H}_{\mathcal{L}}\right), \forall j = 0..\nu-1, f = 0..q-1.$$

Note that the complexity, growing with $O(q^{\kappa})$, becomes prohibitively large if the number of codewords is large. Therefore, *Gallager's algorithm* [34] breaks

Algorithm 1 $SISO_{\text{Exhaustive}}$ Procedure

$\mathbf{P}^{y,Post} = SISO_{\text{Exhaustive}}(\mathbf{P}^{y,Prio}, \mathbf{H}_{\mathcal{L}})$

1. $\Pr(\mathbf{y} \mid \mathbf{H}_{\mathcal{L}}) = \begin{cases} \prod_{j=0}^{\nu-1} \mathbf{P}^{y,Prio}[j, F(y[j])] & \text{if } \mathbf{H}_{\mathcal{L}}\mathbf{y} = 0 \\ 0 & \text{if } \mathbf{H}_{\mathcal{L}}\mathbf{y} \neq 0 \end{cases}$
2. $\mathbf{P}_{j,f}^{y,Post} = \sum_{\mathbf{y}|y[j]=F^{-1}(f)} \Pr(\mathbf{y} \mid \mathbf{H}_{\mathcal{L}}), \forall j = 0 : \nu - 1, f = 0 : q - 1$

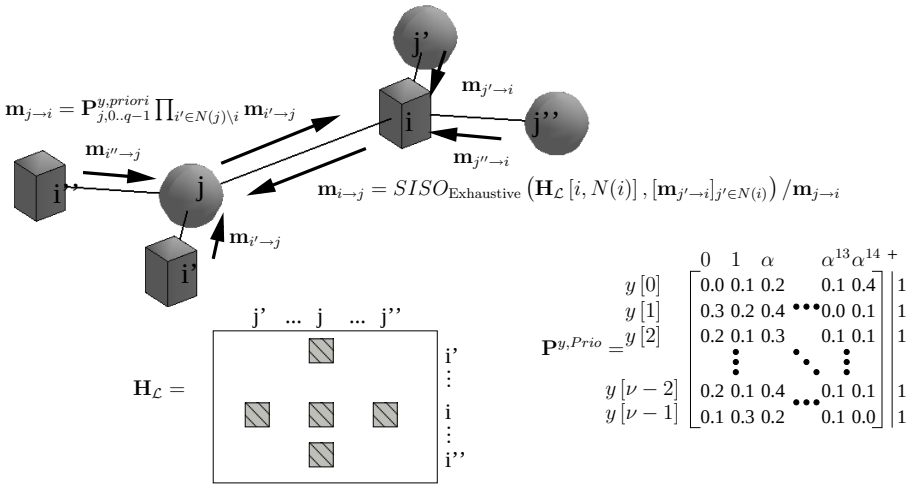


Figure 6.4: Gallager's algorithm illustrated in a Tanner Graph.

down one large code into several smaller ones, where each row of $\mathbf{H}_{\mathcal{L}}$ is considered to be a small (component) code characterized by a parity check matrix that equals this specific row. If row i of $\mathbf{H}_{\mathcal{L}}$ is sparse with row density (number of non-zeros) k_i, the number of codewords for this subcode is only q^{k_i-1}. The complexity then becomes $\sum_i O\left(q^{k_i-1}\right) \approx O\left(q^{\max_i(k_i)-1}\right)$, which can be significantly lower if $\mathbf{H}_{\mathcal{L}}$ is sparse.

Gallager's algorithm is often viewed as a message passing algorithm in a *Tanner graph* (Figure 6.4). The latter is a bipartite undirected graph with $\nu - \kappa$ check nodes (corresponding to each component code, represented by squares) and ν variable nodes (corresponding to the ν codeword symbols, represented by circles). $\mathbf{H}_{\mathcal{L}}$ is viewed as the adjacency matrix of this graph. The algorithm is shown below; it differs from the classical representation that works with *Log Likelihood Ratios* and not with probability vectors. Note in this context that all multiplications and divisions are performed in a component-wise fashion.

$\mathbf{m}_{i\to j}$ is the message ($q \times 1$ probability vector) sent from parity check node i to a variable node j, and can be viewed as the (extrinsic) information that can be learned about j, knowing that all the variable nodes connected to i

Algorithm 2 Gallager's algorithm

$\mathbf{P}^{y,Post} = SISO_{\text{Gallager}}(\mathbf{P}^{y,Prio}, \mathbf{H}_{\mathcal{L}})$

1. Initialization
   $\mathbf{m}_{i\to j} = \left[\frac{1}{q}, ..., \frac{1}{q}\right]$
2. for iter = 1..N_{iter}
   2.a. $\mathbf{m}_{j\to i} = \mathbf{P}^{y,Prio}_{j,0..q-1} \prod_{i' \in N(j)\setminus i} \mathbf{m}_{i'\to j}$
   2.b. $\mathbf{m}_{i\to j} = SISO_{\text{Exhaustive}}\left(\mathbf{H}_{\mathcal{L}}\left(i, N(i)\right), \left[\mathbf{m}_{j'\to i}\right]_{j' \in N(i)}\right) / \mathbf{m}_{j\to i}$
3. $\mathbf{P}^{y,Post}(j, 0:q-1) = \mathbf{P}^{y,Prio}(j, 0:q-1) \prod_{i \in N(j)} \mathbf{m}_{i\to j}$

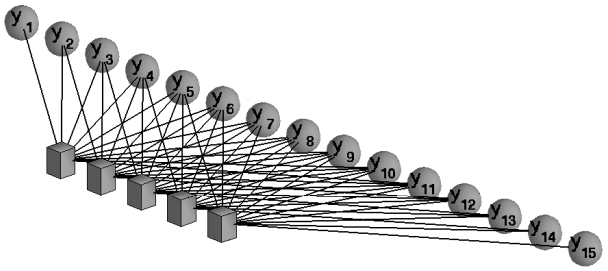


Figure 6.5: Tanner graph corresponding to the (Toeplitz) parity check matrix of $\mathcal{R}\left[15, 10\right]$.

$(N(i))$ should satisfy the check. In step 2.a., all the information gathered about variable node j from all parity checks except i is collected and sent back (message $\mathbf{m}_{j\to i}$) to the parity check i. In step 3 finally, the a posteriori' information of each variable node is found by combining all the messages sent to that node.

Gallager's algorithm suffers from two problems, which are related. The first one is the computational complexity: if a row of $\mathbf{H}_{\mathcal{L}}$ is dense, the number of codewords corresponding to this subcode is too large, resulting in a decoding complexity that is prohibitive. Secondly, a dense $\mathbf{H}_{\mathcal{L}}$ results in cycles in the *Tanner graph*, that can render the algorithm unstable, with a poor performance as a consequence. If the *Tanner Graph* is cycle free however, the exact MAP-estimate is found [50]. For most families of perfect codes, $\mathbf{H}_{\mathcal{L}}$ is dense and as a result, *Gallager's algorithm* will not prove to be very efficient. For instance, in Figure 6.5, the Tanner graph corresponding to the (Toeplitz) parity check matrix of $\mathcal{R}\left[15, 10\right]$ is shown. The row density $k_i = 11$, $\forall i = 0..4$, such that the number of codewords of each subcode equals 16^{10}, which is prohibitively large.

The key concept to overcome this complexity is the clever introduction of auxiliary variables. For example, this is the case when the code is a convolutional code, where the state variables can be introduced to decouple the problem

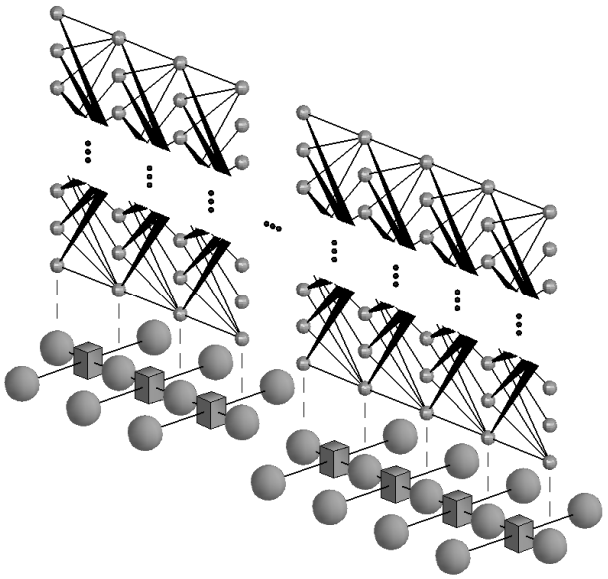


Figure 6.6: Tanner graph corresponding to the trellis representation of $\mathcal{R}\,[15, 10]$. The trellis has 16^5 states, corresponding to subcodes of 16^6 codewords.

(make it sparse). Then Gallager's algorithm can be applied, resulting in BCJR's algorithm [3] which is the SISO version of Viterbi's algorithm. The complexity of BCJR's algorithm is $O\left(q^{\nu-\kappa+1}\right)$. If the filter length $\nu-\kappa+1$ is large, BCJR's algorithm again becomes prohibitively complex. For the example of $\mathcal{R}\,[15, 10]$, the number of codewords in each trellis section equals 16^6, corresponding to a trellis with 16^5 states (see Figure 6.6). The reader is referred to the paper of McEliece *et al.* [50] for further examples.

6.4 Gallager's Algorithm applied to the Filter Bank Decomposition

The insertion of auxiliary variables is exactly the reason why the filter bank decompositions of RS codes are interesting. By introducing the subband variables of a critically subsampled filter bank, the RS code falls apart into smaller codes. Recall the definition of the subband samples $\mathbf{z}$ (Equations 4.39 and 4.40):

$$\mathbf{z} = \mathbf{G}_{\mathcal{D}}\mathbf{x} \tag{6.10}$$

$$\mathbf{y} = \mathbf{G}_{\mathcal{C}}\mathbf{z}\,. \tag{6.11}$$

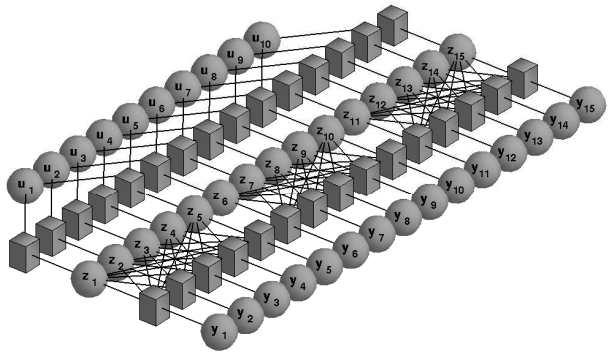


Figure 6.7: Tanner graph corresponding to the filter bank representation of $\mathcal{R}[15, 10]$ (Eq. 6.12).

The code constraint can now be expressed in terms of these subband variables:

$$\begin{bmatrix} \mathbf{I}_\nu & -\mathbf{G}_\mathcal{C} & \\ & \mathbf{I}_\nu & -\mathbf{G}_\mathcal{D} \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ \mathbf{z} \\ \mathbf{u} \end{bmatrix} = \mathbf{0}\,. \tag{6.12}$$

Due to the sparse structure of these matrices, Gallager's algorithm can be successfully applied. The corresponding Tanner graph for the example of $\mathcal{R}[15, 10]$ can be found in Figure 6.7. It has $\nu + \kappa$ parity check nodes and $2\nu + \kappa$ variable nodes, corresponding to the dimensions of the matrix in Eq. 6.12. But a new problem appears: the graph has many small cycles that have a baleful influence on the performance. Due to the block structure of the matrices $\mathbf{G}_\mathcal{C}$ and $\mathbf{G}_\mathcal{D}$, a number of parity check nodes can be grouped into one parity check, thereby removing a lot of small cycles. For example, define a DFT code $\mathcal{C}_F$ by the parity check relation

$$\mathbf{y}[0 : M-1] - \mathbf{G}_\mathcal{C}[0 : M-1, 0 : M-1]\,\mathbf{z}[0 : M-1] = \mathbf{0}\,.$$

Subsequent blocks (length M) of the codeword $\mathbf{y}$ and the subband variables $\mathbf{z}$ are also linked by the same code $\mathcal{C}_F$. Grouping the parity checks corresponding to this code results in the Tanner graph in Figure 6.8.

The parity checks corresponding to the subband codes $\mathcal{D}_m$, $m = 0..M-1$, can also be grouped. The final Tanner graph so obtained is shown in Figure 6.9.

Applying Gallager's algorithm to the latter Tanner graph results in the algorithm which will be discussed in detail. Any a priori information about the dataword can be incorporated in $\mathbf{P}^{u,Prio}$. Otherwise, its entries are $\frac{1}{q}$, indicating equally probable entries. This is also the case for $\mathbf{P}^{z,Prio}$.

The SISO RS decoder operates as follows (Algorithm 3):

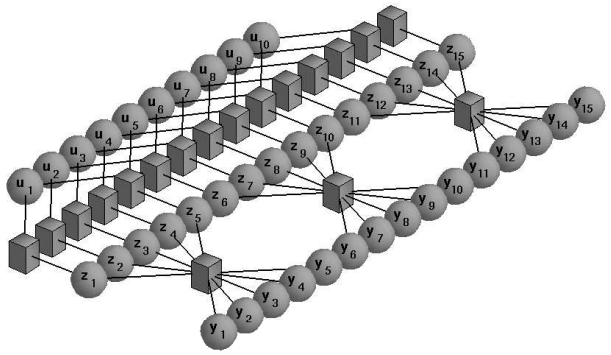


Figure 6.8: Same as Figure 6.7 after grouping the parity checks corresponding to a DFT code.

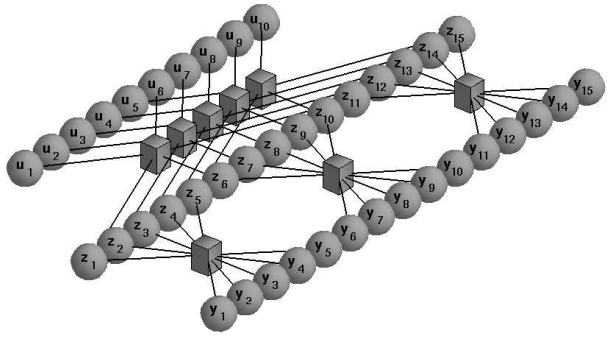


Figure 6.9: Same as Figure 6.8 after grouping the parity checks corresponding to a subband code.

- In the initialization step, the extrinsic information of the subband variables $\mathbf{z}$ is set to $\mathbf{P}^{equi}$ with equally probable entries $\frac{1}{q}$. In step 3, the a posterior probabilities (APP) of the subband variables $\mathbf{z}$ are calculated based on a DFT relation between each consecutive block of M-samples of $\mathbf{z}$ and $\mathbf{y}$:

$$[\mathbf{y}]_{mM+(0:M-1)} = \tilde{\mathbf{F}}_M^{-1}[\mathbf{z}]_{mM+(0:M-1)}, \forall m = 0 : \nu' - 1\,. \tag{6.13}$$

  This operation is performed in parallel by ν' $SISO_{\mathrm{DFT}}$ decoders. If the DFT size is sufficiently small, these decoders operate on the table with all possible codewords (using $SISO_{\mathrm{Exhaustive}}$).

- In step 3.b, the (new) extrinsic information about $\mathbf{z}$ 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 since the (old) extrinsic information does not hold any information: $\mathbf{P}^{z,Ext} = \mathbf{P}^{equi}$.

- This *extrinsic information* $\mathbf{P}^{z,Ext}$ is then fed into M SISO decoders corresponding to the M 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 $\mathbf{u}$ in conjunction with extrinsic information about the subband variables $\mathbf{z}$. Even when there is no a priori information about the dataword, the $SISO_{\mathrm{BCJR}}$ decoder delivers new extrinsic information about $\mathbf{z}$ since $\mathbf{z}$ has redundancy. This extrinsic information is subsequently fed back to the $SISO_{\mathrm{DFT}}$ decoders, which can now deliver APP for the codeword too.

After several Gallager-like iterations[1] (step 2), the probabilities (hopefully) converge. This convergence issue can be found in any Gallager type of algorithm, e.g. in a turbo decoder. It is thoroughly studied since the invention of turbo codes, resulting e.g. in EXIT charts [71]. One of the most important developments in this area is density evolution [58, 19, 14]. With density evolution, it becomes possible to analyze the behavior of Gallager's algorithm under various conditions, like a changing noise model, etc. A detailed EXIT chart analysis of the SISO RS decoder is out of the scope of this dissertation. Nevertheless, simulation results indicate that – on average – convergence settles in after a few iterations.

Let us emphasize that the above algorithm is the result of the application of *Gallager's algorithm* to a simple *Tanner graph* with 3 types of variable nodes ($\mathbf{y}$, $\mathbf{z}$ and $\mathbf{u}$) and 2 types of parity checks ($SISO_{\mathrm{DFT}}$ in between $\mathbf{y}$ and $\mathbf{z}$, $SISO_{\mathrm{BCJR}}$ in between $\mathbf{z}$ and $\mathbf{u}$). Note that, according to *Gallager's algorithm*,

[1] This principle of exchanging information using several iterations is also referred to as the *turbo-principle*.

Algorithm 3 SISO Reed-Solomon decoder

$$\begin{bmatrix}\mathbf{P}^{y,Post}\\ \mathbf{P}^{z,Post}\\ \mathbf{P}^{u,Post}\end{bmatrix} = SISO_{\text{Filterbank}}\left(\begin{bmatrix}\mathbf{P}^{y,Prio}\\ \mathbf{P}^{z,Prio}\\ \mathbf{P}^{u,Prio}\end{bmatrix}, \begin{bmatrix}\mathbf{I}_\nu & -\mathbf{G}_\mathcal{C} & \\ & \mathbf{I}_\kappa & -\mathbf{G}_\mathcal{D}\end{bmatrix}\right)$$

1. Initialization:
   $\mathbf{P}^{z,Ext} = \mathbf{P}^{equi}$
2. While maximum number of iterations not reached
   3. For $n = 0$ to $\nu' - 1$
      3.a. $\begin{bmatrix}\mathbf{P}^{y,Post}[j,f]\\ \mathbf{P}^{z,Post}[j,f]\end{bmatrix} = SISO_{\text{DFT}}\left(\begin{bmatrix}\mathbf{P}^{y,Prio}[j,f]\\ \mathbf{P}^{z,Ext}[j,f]\end{bmatrix}, \begin{bmatrix}\mathbf{I}_M & -\tilde{\mathbf{F}}_M^{-1}\end{bmatrix}\right)$

      3.b. $\mathbf{P}^{z,Ext}[j,f] = \text{Normalize}(\mathbf{P}^{z,Post}[j,f]/\mathbf{P}^{z,Ext}[j,f])$
      with $j = nM + (0:M-1), f = 0:q-1$
   4. For $m = 0$ to $M - 1$
      4.a. $\begin{bmatrix}\mathbf{P}^{z,Post}[j,f]\\ \mathbf{P}^{u,Post}[j',f]\end{bmatrix} = SISO_{\text{BCJR}}\left(\begin{bmatrix}\mathbf{P}^{z,Ext}[j,f]\\ \mathbf{P}^{u,Prio}[j',f]\end{bmatrix}, \begin{bmatrix}\mathbf{I}_{\nu'} & -\mathbf{G}_{\mathcal{D}_m}\end{bmatrix}\right)$

      4.b. $\mathbf{P}^{z,Ext}[j,f] = \text{Normalize}(\mathbf{P}^{z,Post}[\nu,f]/\mathbf{P}^{z,Ext}[\nu,f])$
      with $j = M(0:\nu'-1)+m, j' = M(0:N_m-1)+m, f = 0:q-1$

the SISO decoders only use a priori information about $\mathbf{y}$ and $\mathbf{u}$ since there is never extrinsic information originating from any other check than the SISO decoder under consideration.

The complexity of $SISO_{\text{DFT}}$ and $SISO_{\text{BCJR}}$ grows as $O\left(q^M\right)$, $O\left(q^{\left(\frac{n}{M}-\left\lceil\frac{k}{M}+1\right\rceil\right)}\right)$ respectively. The complexity of the SISO RS decoder grows as follows:

$$O\left(q^{\max\left(M,\frac{\nu}{M}-\left\lceil\frac{\kappa}{M}+1\right\rceil\right)}\right).$$

The complexity is seen to depend on M, the number of bands. If the filter bank techniques are applied recursively, as described in Section 5.4.1, M can be factored as $\prod_i M_i$. This allows us to further reduce the complexity to

$$O\left(q^{\max_i\left(M_i,\frac{\nu}{\prod_i M_i}-\left\lceil\frac{\kappa}{\prod_i M_i}+1\right\rceil\right)}\right).$$

An example of a Tanner graph of a larger $\mathcal{R}[63,54]$ code with $M = 9 = 3\cdot 3$ bands can be found in Figure 6.10. In fact, it is well known that, in a filter application, filter banks offer a significant complexity reduction using fast FIR filtering techniques, as described in detail by Vetterli [78]. An equivalent complexity reduction by the use of filter banks is obtained here for another application, namely in the field of SISO RS decoding.

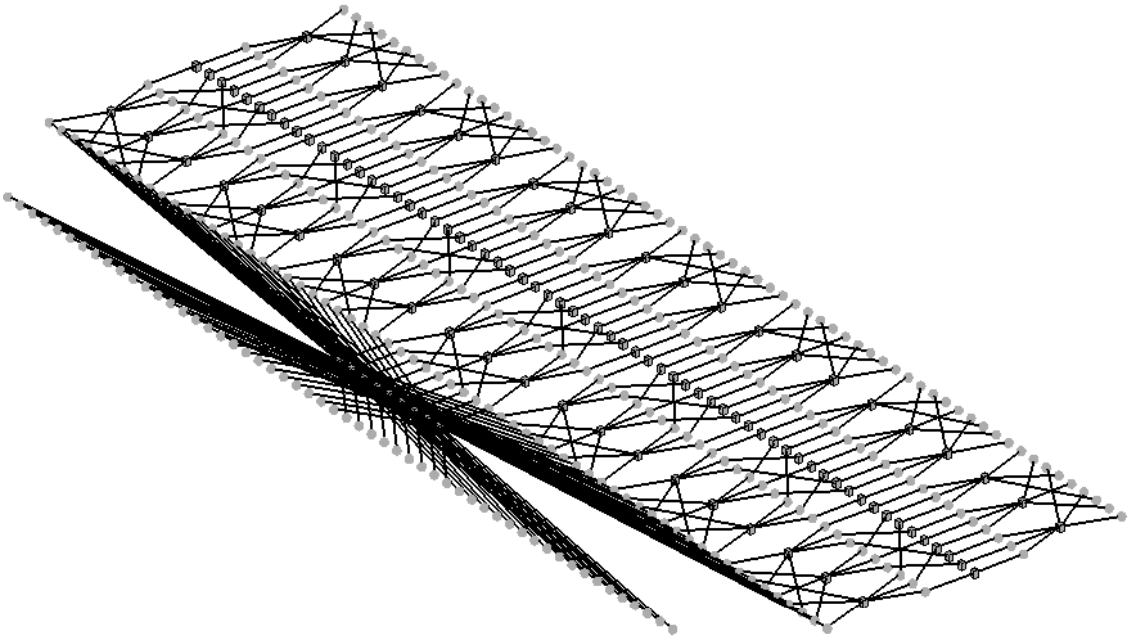


Figure 6.10: Tanner graph of $\mathcal{R}\,[63, 54]$. The corresponding filter bank is obtained by recursively applying the filter bank decomposition as explained in Section 5.4.1.

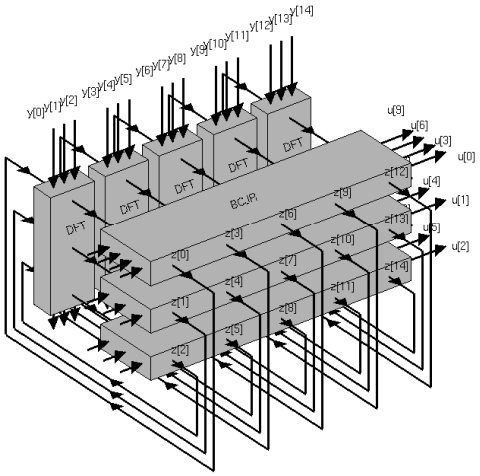


Figure 6.11: Block scheme of SISO decoder for $\mathcal{R}\,[15, 10]$ based on the filter bank shown in Figure 4.2.

Example 6.2 Applying Algorithm 3 to our example of $\mathcal{R}\,[15, 10]$ leads to the block diagram in Figure 6.11. It is based on the filter bank in Figure 4.2. The subband variables $\mathbf{z}$ are used to exchange information between the codeword $\mathbf{y}$ and the dataword $\mathbf{u}$. In each iteration, using the a priori information ($\mathbf{u}$: left; $\mathbf{y}$: top) an update of the a posterior probabilities is returned ($\mathbf{u}$: right; $\mathbf{y}$: bottom).

Also note the interleaving in Figure 6.11. The n-th DFT decoder operates on the variables $\mathbf{z}_{nM+(0:M-1)}$ while the the m-th BCJR decoder is connected with $\mathbf{z}_{M(0:\nu'-1)+m}$. Hence, the information is spread very fast over the graph.

Note that complexity-wise, a brute force application of $SISO_{\text{Exhaustive}}$ operates on a table of 16^{10} possible codewords. Application of $SISO_{\text{BCJR}}$ results in an exhaustive combination and marginalization of a table containing 16^6 codewords, per trellis section. This complexity is further reduced to 16^3 using the $SISO_{\text{Filterbank}}$. △

Before discussing some simulation results, it is important to mention the equivalence with *Turbo codes* [7]. Turbo codes are in fact parallel concatenated convolutional codes (PCCC) where *BCJR's algorithm* is again 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 [50]. 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 now connected by means of

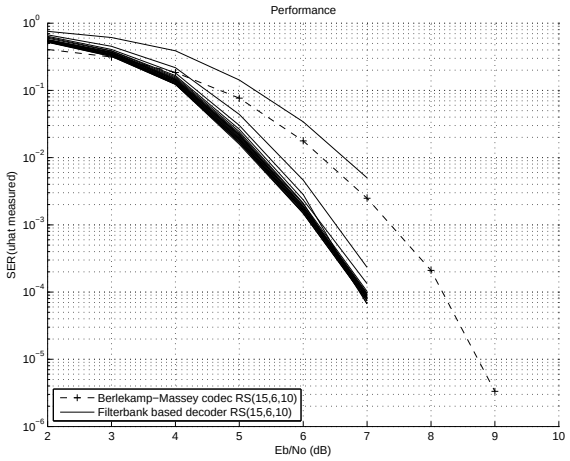


Figure 6.12: Simulation results for $\mathcal{R}\left[15, 6\right]$

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 is an RS code.

In Figure 6.12, simulation results of the $\mathcal{R}\left[15, 6\right]$ 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 $\leq 10^{-4}$, soft decoding using the filter bank based soft decoder provides a coding gain of approximately 1.5 dB compared to a conventional hard decision decoding algorithm, namely *Berlekamp-Massey's algorithm*. Remark also that *Gallager's algorithm* rapidly converges after only a few iterations.

Note that in this section, we only considered decoding $\mathcal{R}^{\Pi_{\nu'}}\left[\nu, \kappa\right]$. It is very hard (from a complexity point of view) to obtain (a MAP estimate of) the corresponding dataword of $\mathcal{R}\left[\nu, \kappa\right]$, since this requires a SISO decoder for the dense matrix V. So at the encoder side, $\mathcal{R}^{\Pi_{\nu'}}\left[\nu, \kappa\right]$ must be used to overcome these problems. Theoretically, this decoding algorithm can also be used when the analysis bank is not a simple serial to parallel converter, e.g. generated by $\mathbf{G}^{V'}_{\mathcal{R}^{\Pi_{\nu'}}}$. In this case however, an extra stage of $SISO_{\mathrm{DFT}}$ decoders must be applied to obtain soft information on the dataword $\mathbf{u}$. In general however, since all the information about $\mathbf{u}$ must be derived from scratch (and ripple through the network) the performance degrades.

Luckily, in practice, systematic RS codes are used in many applications to overcome these problems. Indeed, note that the decoding Algorithm 3 based on the FIR filter bank can be applied without modifications to *any* systematic RS code generated by $\mathbf{G}^{sys}_{\mathcal{R}}$ since the latter is equivalent with $\mathcal{R}^{\Pi_{\nu'}}\left[\nu, \kappa\right]$

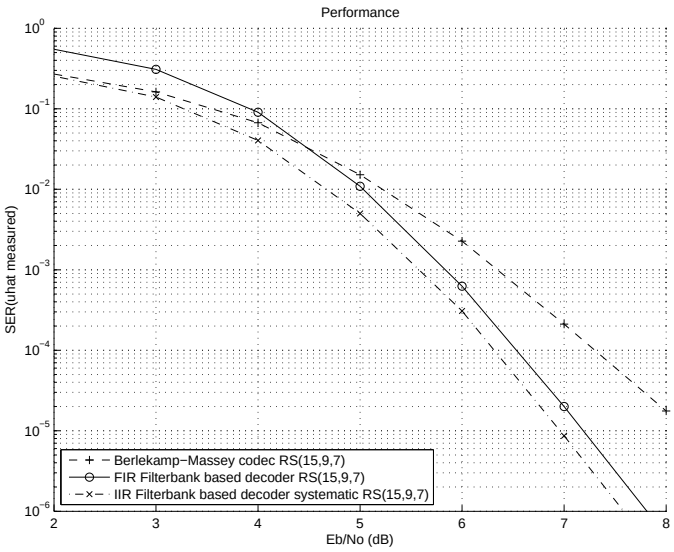


Figure 6.13: Simulation results for $\mathcal{R}\,[15, 9]$

(Section 2.2.4). In the systematic case, soft information about the dataword is readily available in the codeword. This also means that the SISO subband decoders (implemented using BCJR's algorithm) can be simplified in the sense that the computation of $\mathbf{P}^{u,Prio}$ and $\mathbf{P}^{u,Post}$ can be skipped. Finally, since a priori information of the codeword (and thus dataword) is immediately available through the channel, one may expect a better decoding performance (in terms of symbol error rate). The same conclusions hold for the IIR filter bank based decoder. This is confirmed by the simulation results of the $\mathcal{R}\,[15, 9]$ code shown in Figure 6.13. The symbol error rate after the 5-th iteration is shown for the filter bank based SISO decoders. At symbol error rates less than 10^{-5}, the filter bank based RS decoder provides a coding gain of more than 1 dB compared to a conventional hard decision decoding algorithm, namely *Berlekamp-Massey's algorithm*. The systematic variant also performs slightly better than the non-systematic counterpart, as expected.

6.5 Further Research

In this chapter, we have focused on Gallager's algorithm to find the APP in the Tanner graph. As mentioned earlier, this fundamental algorithm suffers from some drawbacks, like divergence due to cycles. This is the major source of performance degradation. Recently, since it is acknowledged that many real world phenomena can be modeled by graphs containing cycles [69], Gal-

lager's algorithm has attracted a lot of interest. Gallager's algorithm in fact has been (re)invented many times in various domains like ECC (Gallager [34], Tanner [70]), Bayesian networks and Belief Propagation (BP) (Pearl [53]), Computer Vision (Geman [35]), Statistical physics (Ising [41]), etc. Consequently, a lot of research is spent to properly understand its behaviour. Applying recent results obtained in these fields as described below is in our opinion the most promising further research direction.

As a first step, EXIT chart [71] analysis can lead to a better understanding of the SISO RS decoder. Another major breakthrough in this field is due to Yedidia [83], which states that the fixed points of a BP algorithm are zero-gradient points of a well defined energy function, called *Bethe free energy*. In this sense, Gallager's algorithm can be seen as an optimization method, and other optimization algorithms can be applied. I.e. modified algorithms can be found that not only have the same fixed points, but provably descend the *Bethe free energy*. Another interesting approach can be found in [69]. In this paper, the inference problem is solved by exactly solving a series of modified estimation problems on (cycle free) spanning trees embedded in the original cyclic graph.

Apart from increasing the decoding performance, the proposed decoding methods should also be tested for longer RS codes, with quantized reliability matrices. If the field size grows, many probability entries become zero, and the (memory) requirements of the algorithm can be further reduced. Also note that, at this time, the exact MAP solution is found for each component code. Other low complexity SISO algorithms can be plugged in as a replacement. As such, the use of the filter bank decomposition is not limited to Gallager type of algorithms.

6.6 Conclusion

This chapter deals with a first application of filter bank decompositions of RS codes, namely SISO RS decoding. The development of an efficient SISO RS decoder is a fundamental open problem in coding theory, that has attracted a lot of attention recently. The filter bank representation developed in Part I is the key concept in finding a suitable decomposition for RS codes. Essentially, the filter bank representations allows us to break one large RS code into a number of subcodes with a sparse interconnect. The subband variables play a central role in this decomposition in order to break the SISO decoder into smaller SISO decoders of the subcodes that exchange information, as dictated by Gallager's algorithm. Simulation results show more than 1 dB of additional decoding gain opposed to Berlekamp-Massey's algorithm.

As a concluding remark, note that the key concept behind the SISO RS decoder

is the sparsity of the matrix decompositions (see Eq. 3.20). If the parity check matrix $\mathbf{H}_{\mathcal{L}}$ of a linear code $\mathcal{L}[\nu, \kappa]$ is sparse, then Gallager's algorithm can be applied. A well known application is SISO decoding of Low Density Parity Check (LDPC) codes. If $\mathbf{H}_{\mathcal{L}}$ is a dense matrix, applying Gallager's algorithm results in a high complexity, together with a poor performance due to cycles in the graph. However, if $\mathbf{G}_{\mathcal{L}}$ is a Toeplitz matrix ($\mathcal{L}[\nu, \kappa]$ is a convolutional code) then the state variables can be added to obtain an overall sparse representation of the code. This is the main idea behind Viterbi's algorithm (or its SISO counterpart: BCJR's algorithm). If the filter then has a large constraint length, there are too many possible states and Viterbi's algorithm becomes impractical again. This is e.g. the case for an RS code. If the code $\mathcal{L}[\nu, \kappa]$ however is a cyclic RS code, it is shown that the subband variables of a critically subsampled filter bank representation can be used in Gallager's algorithm to become an overall sparse structure.

Chapter 7

Reed-Solomon Codes and OFDM

7.1 Introduction

Orthogonal Frequency Division Multiplexing (OFDM) is a well known technique in wireless (e.g. DAB/DVB [1, 2, 72]) as well as in wireline applications, where it is better known as Discrete Multitone (e.g. xDSL [13]). It has several advantages over single carrier systems, e.g. it is resistant against dispersion in the transmission channels, and efficient hardware implementations exist for the equalizer and the DFT based modulation/demodulation. One of the major drawbacks of OFDM is its high Peak to Average Power Ratio (PAPR). This limits the transmission range and requires that the transmit amplifiers have a large input power back-off. If the peak power is limited, the average power allocated must be reduced, thereby limiting the transmission range. Moreover, to allow the transmit amplifiers to operate in their linear region, a large input power back-off is required causing a dramatic increase in power consumption. In many low-cost operations, the disadvantages outweigh all the benefits of OFDM [52].

A second drawback of OFDM is its lack in frequency diversity. To overcome fast fades, coding must be applied. However, in a typical design, the error correcting code is designed separately from the OFDM front-end. This chapter demonstrates that it is not straightforward to design a good error-correcting code that matches with the OFDM-modulator.

This chapter addresses these two problems based on the critically subsampled filter bank representation of Reed-Solomon codes developed in Part I. First,

a brief overview of OFDM is given in Section 7.2. In Section 7.3, the PAPR problem is tackled by presenting the Galois field counterpart of an Orthogonal Frequency Division Multiplexing system, referred to as GF-OFDM. In this scheme, which is merely of theoretical interest, the circulant channel matrix is decomposed into parallel channels by using DFT matrices in a Galois field rather than in the complex field. A modulo operation is added at the receiver to ensure the compatibility between the RS code (in the Galois field) and the channel (in the complex field). This scheme has the advantage of a low Peak to Average Power Ratio (PAPR), since the modulo operation inherent to the Galois field limits the power of the modulated signal.

In a second step (Section 7.4), the combination of the GF-OFDM scheme with Reed-Solomon codes is shown. Relying on the critically subsampled filter bank representation of RS codes developed in Part I, it is seen how RS codes can be merged with GF-OFDM. The overall scheme, referred to as RS-OFDM, shows an error correcting code that is matched to the OFDM-modulator; the optimal Hamming distance of the RS code is preserved by the OFDM modulator and by the channel. This property makes it possible to perform equalization after decoding. Performance-wise, RS-OFDM has a significant performance gain over classical (RS) coded OFDM. However, the modulo operation at the receiver has a baleful influence on the performance. This problem can be overcome by a partial equalization preceding the modulo operation, performed by a whitening matched filter as illustrated in Section 7.4.1.

In the literature, a number of papers can be found describing the relationship between error correcting codes and DFT. In [47], Marshall describes a real valued DFT code, where the generator matrix consists of K rows selected from an M-point DFT matrix in the complex field. All codewords of such a DFT-code will have $M-K$ missing frequency components, corresponding to the $M-K$ columns which were not selected. If $d-1$ missing frequency components are chosen to be cyclically adjacent, the DFT code resembles a (real valued) BCH code with designed distance δ. Complex number BCH codes are then analogous to RS codes. Marshall's approach can be used to construct an OFDM system which transmits zeros on a number of cyclically adjacent bands which are then used for error correction at the receiver. Although the underlying ideas are similar, our approach substantially differs from [47] in many respects, making our methods applicable in practice: in our approach, the DFT size is typically much smaller than the block length of the code. First, redundancy is added to the dataword by a number of independent RS-like subband codes. Their outputs are subsequently fed into a synthesis bank, which essentially takes the (GF) DFT of each block of data symbols just as in a traditional OFDM scenario. Hence, in our approach, the coding aspect (adding redundancy) is completely separated from the modulator, which resembles a classical OFDM modulator. As a second difference, in our approach, all operations are performed in a Galois field of odd characteristic, i.e. $\mathbb{F}_p$ with p a prime number. To be compliant with

the channel operating in the complex field, the isomorphism between $\mathbb{F}_p$ and the integers modulo p will be called upon. By working in the Galois field, the PAPR problem is solved simultaneously.

The idea of jointly solving the PAPR and the error correcting code (ECC) design problem has also appeared in literature. Jones and Wilkinson [43] rely on block codes with codewords corresponding only to these sequences with a low PAPR. In their paper, it is noted that these codes can also be used for error correction, thereby jointly attempting to solve the PAPR as well as the code design. In a similar approach to solve this problem, Davis *et al.* [18] obtained a class of low PAPR with a large minimum distance. These codes are based on cosets of Reed-Muller codes.

Finally, in Section 7.5, some interesting future research directions are described. More specifically, an extension of OFDM towards OFDMA (Orthogonal Frequency Division Multiple Access) is investigated. OFDMA is a special kind of Code Division Multiple Access (CDMA). CDMA offers many desirable features such as dynamic spectrum sharing, narrow-band interference suppression and its ability to cope with the asynchronous nature of multimedia data traffic. Above all, CDMA makes it possible to deliver higher data-rates compared to conventional multiple access techniques such as time-division multiple access (TDMA) and frequency-division multiple access (FDMA) [36]. Therefore, CDMA has been adopted in many standards including the International Mobile Telecommunications (IMT-2000) standard and the Universal Mobile Telecommunications System (UMTS) standard [51]. The broadband character of such cellular wireless communications combined with the high transmission rates cause the inherent time dispersive (frequency selective) character of the channel. In this context, it is not surprising that multi-carrier communication schemes and more specifically OFDM draws a lot of attention [80]. In 1993, as could be expected, new multiple access schemes based on a *combination of code division and OFDM* appeared. These schemes and their variations are known under acronyms as MC-CDMA [84, 24, 12], MC-DS-CDMA [17], MT-CDMA [76]. These schemes show a spreading (in the time or in the frequency domain) followed by an OFDM modulator. The orthogonality among the users is preserved even through a convolutive channel, making these schemes multi-user interference (MUI) resilient. Another variant, called OFDMA, was originally proposed in a Cable-TV context [60]. In its simplest form the spreading code of an OFDMA user corresponds to one column of the OFDM DFT matrix. Therefore, it can be seen as OFDM's most straightforward counterpart for multi-access schemes. Hence, OFDMA as well as the other MC-CDMA schemes inherit OFDM's advantages as well as its disadvantages as discussed above. To increase robustness against channel fades, error correcting coding [60] or frequency hopping [85] is applied. Typically, error correcting codes are applied on a per user basis, without questioning the performance (distance) of the overall system. This is illustrated in [63], in which the authors attempt to find an

expression for the Euclidean Distance of a CDMA system. At the end of this chapter, we will show how the filter bank representation of RS codes can be used to extend the RS-OFDM system to support multiple users.

7.2 OFDM System Model

The core of the OFDM architecture is a block based system, processing data in a block-wise fashion. A data block $\mathbf{z}$ is modulated using the inverse M-point discrete Fourier transform (IDFT), obtaining $\mathbf{y}$, and is prepended with a cyclic prefix of length L. The multi-path propagation channel is modeled as a finite impulse response filter $h(z^{-1}) = h[0] + h[1]\,z^{-1} + h[2]\,z^{-2} + \dots$ with noise added. Assuming perfect symbol and block synchronization and that the channel order is not larger than L, inter-block interference is avoided such that $\tilde{\mathbf{y}}$ is received which is specified as follows:

$$\begin{aligned} \mathbf{y} &= \mathbf{F}_M^{-1}\mathbf{z} && (7.1)\\ \tilde{\mathbf{y}} &= \underbrace{\left[\mathbf{O}_{M,L} \mid \mathbf{I}_M\right]\mathbf{H}\left[\begin{array}{c}\mathbf{I}_L \mid \mathbf{0}_{L,M-L}\\ \hline \mathbf{I}_M\end{array}\right]}_{\tilde{\mathbf{H}}}\mathbf{y} + \mathbf{n} && (7.2)\\ \tilde{\mathbf{z}} &= \mathbf{F}_M\tilde{\mathbf{y}}\,. && (7.3) \end{aligned}$$

In these equations, $\mathbf{F}_M$ represents a (complex field) $M \times M$ DFT matrix with $\mathbf{F}_M[i,j] = W_M^{ij}$ and W_M is an M-th root of unity: $W_M = e^{2\pi j/M}$. The channel matrix $\mathbf{H}$ is defined as

$$\mathbf{H} = \begin{bmatrix} h[0] & & & & & & \\ h[1] & \ddots & & & & & \\ \vdots & \ddots & \ddots & & & & \\ h[L] & & \ddots & \ddots & & & \\ & \ddots & & \ddots & \ddots & & \\ & & h[L] & \dots & h[1] & h[0] \end{bmatrix} \qquad (7.4)$$

The cyclic prefix (CP) insertion and removal transforms the channel matrix $\mathbf{H}$ into a circulant matrix $\tilde{\mathbf{H}}$ which is diagonalized by $\mathbf{F}_M$ and $\mathbf{F}_M^{-1}$, i.e.

$$\tilde{\mathbf{E}} = \mathbf{F}_M\tilde{\mathbf{H}}\mathbf{F}_M^{-1}\,. \qquad (7.5)$$

Hence, an estimate of $\mathbf{z}$ can be obtained as

$$\hat{\mathbf{z}} = \tilde{\mathbf{E}}^{-1}\tilde{\mathbf{z}} = \mathbf{z} + \tilde{\mathbf{E}}^{-1}\mathbf{F}_M\mathbf{n}\,.$$

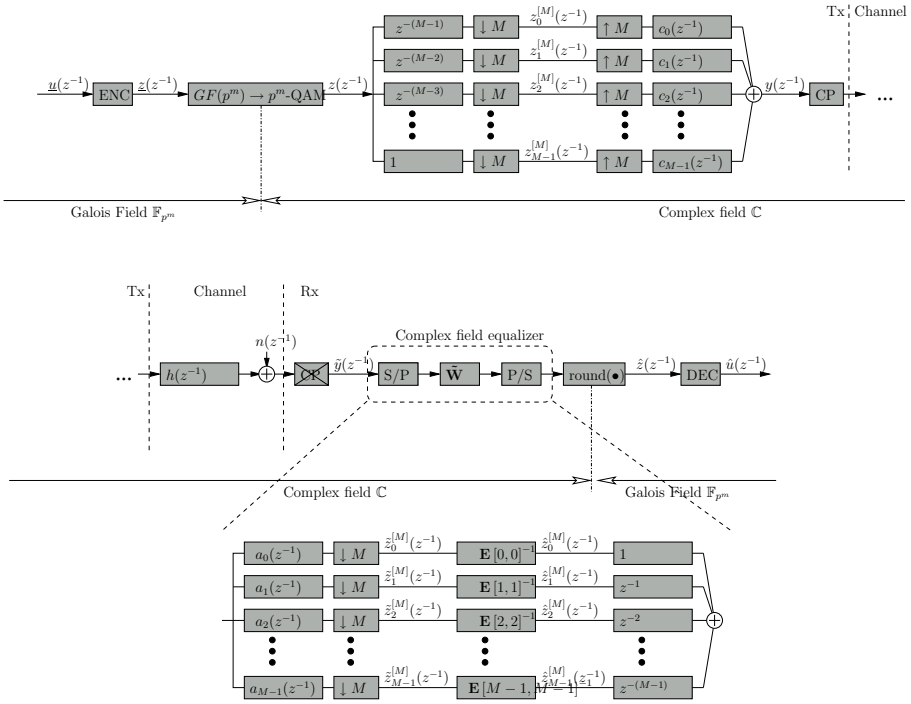


Figure 7.1: Transmultiplexer structure of a baseband discrete-time model of an OFDM system.

Considering the transmission of multiple blocks, the baseband discrete-time model of an OFDM system is seen as a transmultiplexer structure in Figure 7.1. This representation makes it easier to establish the link with the filter banks further on. Moreover, the polyphase decomposition allows to properly describe the block based system. Now, an input sequence $z(z^{-1}) = z[0] + z[1]\,z^{-1} + z[2]\,z^{-2} + \ldots$ is written as a block of M samples using its M-fold polyphase components:

$$\mathbf{z}_l^{[M]}(z^{-1}) = \sum_{k=0}^{\infty} z[l + kM]\,z^{-k}\,. \tag{7.6}$$

These polyphase components can be stacked into a vector

$$\mathbf{z}^{[\mathbf{M}]}(z^{-1}) = \left[z_0^{[M]}(z^{-1}) \quad z_1^{[M]}(z^{-1}) \quad \ldots \quad z_{M-1}^{[M]}(z^{-1})\right]^T\,. \tag{7.7}$$

Note that in the case of a single transmitted block, all polyphase components are scalars again and $\mathbf{z}^{[\mathbf{M}]} = \mathbf{z} = \left[z[0] \quad z[1] \quad \ldots \quad z[M-1]\right]^T$, just as before. Equations 7.1- 7.3 can now be rewritten using their polyphase components by simply replacing, e.g., $\mathbf{z}$ by $\mathbf{z}^{[\mathbf{M}]}(z^{-1})$. To satisfy $\mathbf{y}^{[\mathbf{M}]}(z^{-1}) = \mathbf{F}_M^{-1}\mathbf{z}^{[\mathbf{M}]}(z^{-1})$ (Eq. 7.1) and $\tilde{\mathbf{z}}^{[\mathbf{M}]}(z^{-1}) = \mathbf{F}_M\tilde{\mathbf{y}}^{[\mathbf{M}]}(z^{-1})$ (Eq. 7.3), the synthesis (analysis) filters need to be defined as

$$c_m(z^{-1}) = \sum_{m'=0}^{M-1} W_M^{-mm'} z^{-m'} \tag{7.8}$$

$$a_m(z^{-1}) = \sum_{m'=0}^{M-1} W_M^{mm'} z^{-m'}\,. \tag{7.9}$$

7.3 Galois Field OFDM

If the channel is a Galois field channel, it can be similarly diagonalized if the M-th root of unity (defining the DFT-matrix) is re-defined accordingly:

$$W_M = \alpha^{\frac{p^m-1}{M}}\,. \tag{7.10}$$

In this equation, α is a primitive $(p^m - 1)$-th root of unity in $\mathbb{F}_{p^m}$. Also note that the M-th root of unity (and thus the DFT matrix) only exists if M divides $p^m - 1$.

The channel is however always defined in the complex or real field. In order to simplify the mathematical description, let us start with a real field channel, which is assumed to be quantized (integer channel coefficients). This assumption will be relaxed in Sections 7.4.1 and 7.4.2.

7.3.1 Integer Channel (PAM-constellations)

If a Galois field of odd characteristic ($\mathbb{F}_p$ with p prime) is chosen, and if a modular reduction is applied at the receiver, the diagonalization of an integer channel (i.e. with integer coefficients) can indeed be obtained with finite field DFT matrices. This is due to the well known fact that there exists an isomorphism between the elements of $\mathbb{F}_p$ and the integers modulo p. Therefore, an addition (multiplication) performed by the channel (in the integer field) can be seen as an addition (multiplication) in $\mathbb{F}_p$ if a modular reduction is introduced at the output of the channel. In the rest of the section, this is explained in more detail.

Let s_0 be a multiplicative generator of the integers modulo p. The previously mentioned isomorphism maps s_0 onto α, the primitive $p-1$-st root of unity in $\mathbb{F}_p$.[1] The complete mapping (isomorphism) between the elements of the Galois field and the integers modulo p:

$$\underline{a} = \alpha^i \rightarrow a = (s_0)^i \bmod p\,. \tag{7.11}$$

From a notational point of view, note that the Galois field representation of a (matrix) variable a is denoted by $\underline{a}$. To limit the power of the signal, the integers are preferably bounded between $-\frac{p-1}{2} \ldots \frac{p-1}{2}$ instead of $0 \ldots p-1$, such that a p-PAM constellation is obtained. Hence, the modulo operation $x \bmod p$ is replaced by $(x)_p$:

$$(x)_p = \text{round}\left(x - p\left\lfloor \frac{(x+\frac{p}{2})}{p} \right\rfloor\right) \tag{7.12}$$

and

$$\underline{a} = \alpha^i \rightarrow a = ((s_0)^i)_p\,. \tag{7.13}$$

Note that x in Eq. 7.12 can now also be non-integer, which will be used further on.

The integer channel coefficients define the matrix $\tilde{\mathbf{H}}$ which can be diagonalized in the Galois field:

$$\underline{\tilde{\mathbf{E}}} = \underline{\mathbf{F}}_M \underline{\tilde{\mathbf{H}}} \underline{\mathbf{F}}_M^{-1}\,. \tag{7.14}$$

This decomposition can then be used to make an alternative OFDM scheme:

$$\underline{\mathbf{y}}^{[\mathbf{M}]}(z^{-1}) = \underline{\mathbf{F}}_M^{-1} \underline{\mathbf{z}}^{[\mathbf{M}]}(z^{-1}) \tag{7.15}$$

$$\tilde{\mathbf{y}}^{[\mathbf{M}]}(z^{-1}) = \left(\tilde{\mathbf{H}}\mathbf{y}^{[\mathbf{M}]}(z^{-1}) + \mathbf{n}(z^{-1})\right)_p \tag{7.16}$$

$$\underline{\tilde{\mathbf{z}}}^{[\mathbf{M}]}(z^{-1}) = \underline{\mathbf{F}}_M \underline{\tilde{\mathbf{y}}}^{[\mathbf{M}]}(z^{-1}) \tag{7.17}$$

$$\hat{\underline{\mathbf{z}}}^{[\mathbf{M}]}(z^{-1}) = \underline{\tilde{\mathbf{E}}}^{-1} \underline{\tilde{\mathbf{z}}}^{[\mathbf{M}]}(z^{-1})\,. \tag{7.18}$$

[1] The corresponding primitive polynomial defining the Galois field $\mathbb{F}_p$ is then $\mathscr{P}(x) = x + (-s_0 \bmod p)$.

In the above equations, all operations are Galois field operations, except Eq. 7.16 which uses the integer representation $\mathbf{y}^{[M]}(z^{-1})$ as input to the channel. Note that these Galois field operations (modulo operations) ensure the low PAPR of the modulated signal, similar to the modulo operation in a dirty paper coding scheme. After taking the modulo-like operation, the result $\tilde{\mathbf{y}}^{[\mathbf{M}]}(z^{-1})$ is returned to the Galois field for demodulation (Eq. 7.17) and equalization (Eq. 7.18). The Galois field counterpart of Eq. 7.16 is

$$\underline{\tilde{\mathbf{y}}}^{[\mathbf{M}]}(z^{-1}) = \underline{\tilde{\mathbf{H}}}\underline{\mathbf{y}}^{[M]}(z^{-1}) + \underline{\mathbf{n}'}(z^{-1}) \tag{7.19}$$

with $\mathbf{n}'(z^{-1}) = (\mathbf{n}(z^{-1}))_p$. The effect of this rounding is important, and will be discussed in detail later on.

Example 7.1 For the sake of convenience, a small Galois field $\mathbb{F}_{11}$ is chosen ($p = 11$) with primitive polynomial $\mathscr{P}(x) = 9 + x$. The primitive $p-1$-th root of unity α is a root of this primitive polynomial:

$$\alpha = -9 \bmod 11 = 2 \tag{7.20}$$

The isomorphism between the elements of the Galois field and the 11-PAM constellation can now be constructed:

$$\begin{array}{cccccccccccc} 0 & 1 & \alpha^1 & \alpha^2 & \alpha^3 & \alpha^4 & \alpha^5 & \alpha^6 & \alpha^7 & \alpha^8 & \alpha^9 \\ 0 & 1 & 2 & 4 & -3 & 5 & -1 & -2 & -4 & 3 & -5 \end{array}$$

Choosing $M = 5$ ($M|p-1 = 10$), the DFT-matrix and its inverse are

$$\underline{\mathbf{F}}_5 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & \alpha^2 & \alpha^4 & \alpha^6 & \alpha^8 \\ 1 & \alpha^4 & \alpha^8 & \alpha^2 & \alpha^6 \\ 1 & \alpha^6 & \alpha^2 & \alpha^8 & \alpha^4 \\ 1 & \alpha^8 & \alpha^6 & \alpha^4 & \alpha^2 \end{bmatrix} \quad \mathbf{F}_5 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 4 & 5 & -2 & 3 \\ 1 & 5 & 3 & 4 & -2 \\ 1 & -2 & 4 & 3 & 5 \\ 1 & 3 & -2 & 5 & 4 \end{bmatrix} \tag{7.21}$$

$$\mathbf{F}_5^{-1} = \frac{1}{\sum_1^M \alpha^0} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & \alpha^8 & \alpha^6 & \alpha^4 & \alpha^2 \\ 1 & \alpha^6 & \alpha^2 & \alpha^8 & \alpha^4 \\ 1 & \alpha^4 & \alpha^8 & \alpha^2 & \alpha^6 \\ 1 & \alpha^2 & \alpha^4 & \alpha^6 & \alpha^8 \end{bmatrix} = \begin{bmatrix} -2 & -2 & -2 & -2 & -2 \\ -2 & 5 & 4 & 1 & 3 \\ -2 & 4 & 3 & 5 & 1 \\ -2 & 1 & 5 & 3 & 4 \\ -2 & 3 & 1 & 4 & 5 \end{bmatrix}. \tag{7.22}$$

A circulant channel matrix $\tilde{\mathbf{H}}$ can be decomposed as follows:

$$\left(\mathbf{F}_5 \underbrace{\begin{bmatrix} 1 & 0 & 0 & -2 & 4 \\ 4 & 1 & 0 & 0 & -2 \\ -2 & 4 & 1 & 0 & 0 \\ 0 & -2 & 4 & 1 & 0 \\ 0 & 0 & -2 & 4 & 1 \end{bmatrix}}_{\tilde{\mathbf{H}}} \mathbf{F}_5^{-1} \right)_{11} = \underbrace{\begin{bmatrix} 3 & & & & \\ & -4 & & & \\ & & 4 & & \\ & & & -4 & \\ & & & & -5 \end{bmatrix}}_{\tilde{\mathbf{E}}}$$

△

7.3.2 Complex Field Channel (QAM constellation)

So far, a Galois field of order p was used, which was shown to be isomorphic to a p-PAM constellation $-\frac{p-1}{2}..\frac{p-1}{2}$. It is well known that QAM constellations have a better performance due to an extra degree of freedom. The goal of this section is to extend the GF-OFDM system of the previous section towards QAM-constellations. More specifically, an isomorphism between the Galois field and QAM constellations is aimed for.

In Section 2.1, it is explained that $\mathbb{F}_{p^m}$ (also denoted as $\mathbb{F}_p^m$) is an extension field of $\mathbb{F}_p$. Any member of $\mathbb{F}_{p^m}$ can be written as an m-tuple of elements in $\mathbb{F}_p$, or alternatively as a polynomial of degree $(m-1)$ with coefficients in $\mathbb{F}_p$. This polynomial representation is useful to define the multiplication in $\mathbb{F}_{p^m}$. Indeed, by defining an irreducible polynomial $\mathscr{P}(x)$, the elements of $\mathbb{F}_{p^m}$ can be described as residue classes of the polynomials with coefficients from $\mathbb{F}_p$, reduced modulo $\mathscr{P}(x)$. All the nonzero elements of $\mathbb{F}_{p^m}$ form a cyclic group of order p^m-1. I.e. any nonzero element can be written uniquely as a power of $\beta \in \mathbb{F}_{p^m}$, with β the residue class of the element $[0,1,0,\ldots,0]$. If $\mathscr{P}(x)$ is also primitive, β is a root of $\mathscr{P}(x)$.

Equivalently, the field of complex numbers is an extension field of the field of real numbers $\mathbb{R}$, denoted as $\mathbb{C} = \mathbb{R}^2$. $\mathbb{R}$ is referred to as the base field. Any member of $\mathbb{C}$ can be written as a 2-tuple of elements in $\mathbb{R}$, or a polynomial with coefficients in $\mathbb{R}$:

$$[a,b] \leftrightarrow a + bx \tag{7.23}$$

Addition in $\mathbb{C}$ can simply be defined as addition of the polynomials. To define multiplication, an irreducible polynomial $\mathscr{P}(x) = 1 + x^2$ with coefficients in $\mathbb{R}$ is specified. Multiplication can now be defined modulo $\mathscr{P}(x)$. This leads to the following (well-known) arithmetic:

$$[a,b] \cdot [c,d] = [a \cdot c - b \cdot d, a \cdot d + b \cdot c] \tag{7.24}$$

$$[a,b] + [c,d] = [a + c, b + d] \tag{7.25}$$

Defining $j \in \mathbb{C}$ as the residue class of the element $[0,1]$, it follows that $\mathscr{P}(j) = 0$ or $j^2 = -1$. Hence, this 'special' element $j \in \mathbb{C}$, $j \notin \mathbb{R}$ is often used in the polynomial representation leading to the notation $a+bj$ instead of $a+bx$. Note that other representations are possible, e.g. multiplication is easier defined by choosing an exponential representation $a + jb = re^{j\theta}$. The 3 representations are summarized in Table 7.1.

Table 7.1: Three representations of elements in $\mathbb{C}$.

2-tuple	polynomial	exponential
$[a,b]$	$a+jb$	$re^{j\theta}$

Unfortunately, the problem is that the primitive polynomial in $\mathbb{C}$ $\mathscr{P}(x) = x^2+1$ is typically *not* a primitive polynomial in $\mathbb{F}_{p^2}$.

Example 7.2 In $\mathbb{F}_{3^2}$ (defined by $\mathscr{P}(x) = x^2 + 2x + 2$, see Example 2.2), the roots of x^2+1 are β^2 and β^6 and these do not generate $\mathbb{F}_{3^2}$. Therefore, another representation of $\mathbb{F}_{3^2}$ is needed to construct the isomorphism between $\mathbb{F}_{3^2}$ and a 9-QAM constellation. △

Again, the trace is called upon to change the basis (see Section 4.3). A special basis transformation unveils the isomorphism between the field $\mathbb{F}_{p^2}$ and a p^2-QAM constellation:

Theorem 7.1 *If the elements* 1 *and* β^n *are linearly independent, there exists an isomorphism between the field* $\mathbb{F}_{p^2}$ *and a* p^2*-QAM constellation, which is defined by the basis* $\boldsymbol{\lambda} = \begin{bmatrix} 1 & \beta^n \end{bmatrix}$, $n = \frac{p^2-1}{4}$.

Proof: This can be shown by proving that $\beta^{n\,2} = -1\lambda\,[0] + 0\lambda\,[1]$. Since $\beta^{2n} \cdot \beta^{2n} = \beta^{p^2-1} = 1$ and $n \neq 0$, β^{2n} must be -1. Note that (p^2-1) is always a multiple of 4 since $p^2 - 1 = (p-1)(p+1)$ and both factors are even (since p is prime). □

Example 7.3 We continue our example of $\mathbb{F}_{3^2}$ with a primitive polynomial in $\mathbb{F}_3$

$$\mathscr{P}(x) = x^2 + 2x + 2 \tag{7.26}$$

With a new basis $\boldsymbol{\lambda} = \begin{bmatrix} 1 & \beta^2 \end{bmatrix}$, the field $\mathbb{F}_{3^2}$ is shown in Table 7.2.

Table 7.2: Isomorphism between elements of $\mathbb{F}_{3^2}$ and elements of 9-QAM $\subset \mathbb{C}$.

2-tuple	polynomial	exponential	element in 9-QAM
$\begin{bmatrix} 0 & 0 \end{bmatrix}$	0	0	0
$\begin{bmatrix} 1 & 0 \end{bmatrix}$	1	1	1
$\begin{bmatrix} 2 & 1 \end{bmatrix}$	$2 + 1\beta^2$	β	$-1 + j$
$\begin{bmatrix} 0 & 1 \end{bmatrix}$	β^2	β^2	j
$\begin{bmatrix} 2 & 2 \end{bmatrix}$	$2 + 2\beta^2$	β^3	$-1 - j$
$\begin{bmatrix} 2 & 0 \end{bmatrix}$	2	β^4	-1
$\begin{bmatrix} 1 & 2 \end{bmatrix}$	$1 + 2\beta^2$	β^5	$1 - j$
$\begin{bmatrix} 0 & 2 \end{bmatrix}$	$2\beta^2$	β^6	$-j$
$\begin{bmatrix} 1 & 1 \end{bmatrix}$	$1 + \beta^2$	β^7	$1 + j$

Note that $\beta^4 = -1$. Hence, multiplication/addition of elements using the (new) polynomial representation is completely equivalent to multiplication/addition in the complex field ($\beta^2 = j$). For example,

$$(2 + j) \cdot (1 + j) = 1 \leftrightarrow \beta \cdot \beta^7 = 1\,. \tag{7.27}$$

Note that all fields of order p^m are isomorphic (Section 2.1). We want to emphasize that the representations are different, but there is a one-to-one mapping between the representations preserving addition and multiplication. Finally, note

that the elements of the base field can better be represented by $-\frac{p-1}{2}..\frac{p-1}{2}$ to limit the energy of the constellation. I.e. addition and multiplication are calculated using $(\cdot)_p$ instead of modulo p.

△

Note that the isomorphism does not exist whenever the elements of the basis $\begin{bmatrix}1 & \beta^n\end{bmatrix}$ are linearly dependent.

Example 7.4 For example, in $\mathbb{F}_{5^2}$, $n = 6$ and hence $\boldsymbol{\lambda} = \begin{bmatrix}1 & \beta^6\end{bmatrix}$. However, since $\beta^0 + \beta^0 = \beta^6$, this is not a valid basis and the isomorphism does not exist. △

It can be proven that the isomorphism exists if and only if $p+1$ does not divide n.

7.4 Combining RS and OFDM:RS-OFDM

Error correcting codes are typically applied around the core (GF-)OFDM system as shown in Figure 7.1. Typically, the optimality of the overall system (concatenation of ECC with the OFDM modulator) is never questioned. Often, RS codes are chosen, e.g. in ADSL since they are optimal for impulse noise correction given their maximal Hamming distance. However, the concatenation of an RS code and the OFDM modulator is seen to lower the Hamming distance of the overall code. Let us illustrate this in the following example.

Example 7.5 Since Galois fields of prime characteristic are of special interest to us ($m = 1$), $\mathcal{R}[10, 5, 6]$ in $\mathbb{F}_{11}$ is chosen. Its roots are chosen $\{\alpha^k\}_{k=0\ldots4}$. Hence, each codeword is a multiple of the generator polynomial

$$\underline{g}(z^{-1}) = \prod_{k=0}^{4}(z^{-1} - \alpha^k)\,. \tag{7.28}$$

Consider a typical concatenated scheme of a (non-systematic) $\mathcal{R}[10, 5, 6]$ followed by a GF-OFDM modulator. The overall Hamming distance of the concatenated scheme can be calculated to be 2, which is a factor 3 less than the original Hamming distance aimed for. Consequently, the error correcting capability of the RS code is drastically reduced. △

Due to the filter bank representation of RS codes, a new scheme can be developed where the DFT synthesis bank of the GF-OFDM modulator is part of

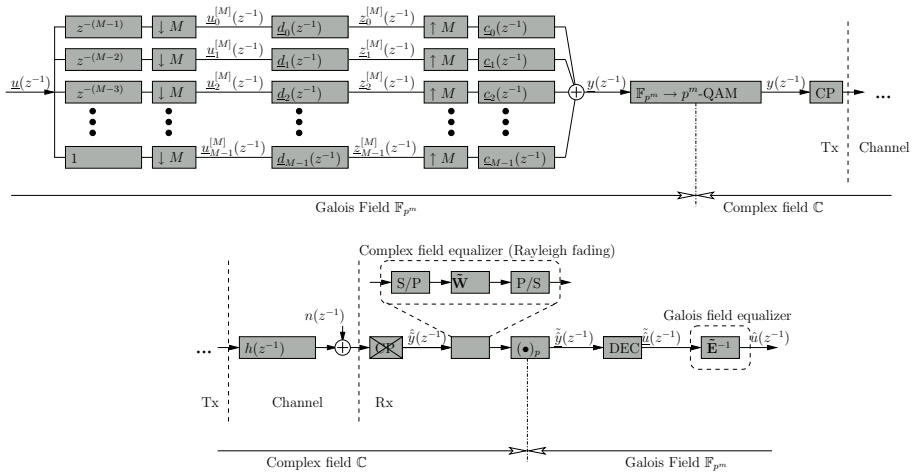


Figure 7.2: Transmultiplexer structure of a baseband discrete-time model of an RS-OFDM system.

the RS code. Hence, the two schemes can be merged, leading to a new scheme referred to as RS-OFDM. Mathematically, the operations performed by the filter bank can be rewritten using polyphase components as

$$\underline{\mathbf{y}}^{[\mathbf{M}]}(z^{-1}) = \underline{\mathbf{F}}_M^{-1} \mathrm{Diag}\left(\underline{\mathbf{d}}(z^{-M})\right) \underline{\mathbf{u}}^{[\mathbf{M}]}(z^{-1})\,. \tag{7.29}$$

The latter equation together with Eq. 7.15 can be merged leading to

$$\tilde{\underline{\mathbf{y}}}^{[\mathbf{M}]}(z^{-1}) = \underline{\tilde{\mathbf{H}}\mathbf{F}}_M^{-1} \mathrm{Diag}\left(\underline{\mathbf{d}}(z^{-1})\right) \underline{\mathbf{u}}^{[\mathbf{M}]}(z^{-1}) + \underline{\mathbf{n}'}(z^{-1})\,. \tag{7.30}$$

In RS-OFDM using $\mathcal{R}\,[\nu, \kappa]$, redundancy is added by multiplying the polyphase components of a dataword $u(z^{-1})$ with $\underline{\mathbf{d}}(z^{-M})$. The resulting subband variables $x_m^{[M]}(z^{-1})$ are sent into the DFT synthesis bank, which is now common to the RS code and the OFDM modulator. This DFT is crucial to the RS code (to obtain a larger distance) and at the same time ensures the diagonalization of the channel as explained in the previous section. Instead of concatenating an RS code with an OFDM modulator, the overall Hamming distance of the RS code is preserved by the RS-OFDM system.

Moreover, it can be shown that also an (integer) channel does not reduce the Hamming distance of the code, which is remarkable. Recalling Eq. 7.30

$$\begin{aligned}
\tilde{\underline{\mathbf{y}}}^{[\mathbf{M}]}(z^{-1}) &= \underline{\tilde{\mathbf{H}}\mathbf{F}}_M^{-1} \mathrm{Diag}\left(\underline{\mathbf{d}}(z^{-M})\right) \underline{\mathbf{u}}^{[\mathbf{M}]}(z^{-1}) + \underline{\mathbf{n}'}(z^{-1}) && (7.31)\\
&= \underline{\mathbf{F}}_M^{-1} \tilde{\underline{\mathbf{E}}} \mathrm{Diag}\left(\underline{\mathbf{d}}(z^{-M})\right) \underline{\mathbf{u}}^{[\mathbf{M}]}(z^{-1}) + \underline{\mathbf{n}'}(z^{-1}) && (7.32)\\
&= \underline{\mathbf{F}}_M^{-1} \mathrm{Diag}\left(\underline{\mathbf{d}}(z^{-M})\right) \underbrace{\tilde{\underline{\mathbf{E}}}\underline{\mathbf{u}}^{[\mathbf{M}]}(z^{-1})}_{\tilde{\underline{\mathbf{u}}}^{[\mathbf{M}]}(z^{-1})} + \underline{\mathbf{n}'}(z^{-1}) && (7.33)
\end{aligned}$$

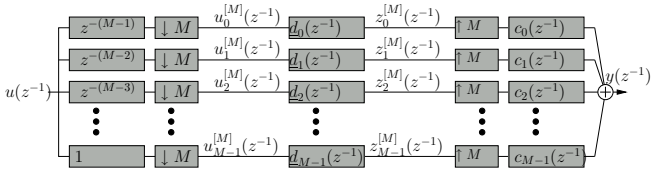


Figure 7.3: Critically subsampled filter bank with M bands. According to Theorem 4.2, each RS code can be represented as such a filter bank. Note the correspondence with the synthesis bank in Figure 7.1.

The last equation shows that the same $\tilde{\mathbf{y}}^{[\mathbf{M}]}$ can be obtained by encoding a *different* dataword $\underline{\tilde{\mathbf{u}}}^{[\mathbf{M}]}$ instead of $\underline{\mathbf{u}}^{[\mathbf{M}]}$. In other words, the channel output is also a codeword (up to the noise). This means that at the receiver, one can first e.g. use *Berlekamp-Massey's algorithm* [49] to find an approximate ML-solution for $\underline{\tilde{\mathbf{u}}}^{[\mathbf{M}]}$, and then perform an equalization on the data according to

$$\underline{\tilde{\mathbf{E}}\mathbf{u}}^{[\mathbf{M}]}(z^{-1}) = \underline{\tilde{\mathbf{u}}}^{[\mathbf{M}]}(z^{-1})\,. \tag{7.34}$$

In this case, equalization and decoding are separated without compromising optimality (no joint decoding-equalization necessary) and can even be swapped.

7.4.1 RS-OFDM with a Serial Zero Forcing Equalizer Design

The previous sections were limited to integer channels, seriously limiting the practical usability of the RS-OFDM scheme. In this section, it is explained how the RS-OFDM scheme is modified to overcome this problem. The difficulty resides in Eq. 7.16, where a decision is taken prior to further equalization in the receiver. There are three possible scenarios regarding equalization (Figure 7.4).

- The first scenario (Eq. 7.17) is depicted in Figure 7.4 with $w(z^{-1}) = 1$ such that $h'(z^{-1}) = h(z^{-1})$. The decision device then causes some information to be lost. For example, in the simplest case of a flat fading channel with a channel gain of $\frac{p-1}{2}$, a noise value larger than 0.5 causes a symbol error to be made. However, one would expect to tolerate noise as large as $\frac{p-1}{4}$. In conclusion, taking decisions prior to equalization is not always a good option.

- In a second scenario e.g. as in a classical OFDM scheme, larger noise can be tolerated with an equalizer in the complex field which would downscale the noise by a factor of $\frac{p-1}{2}$. If the equalizer is designed according to the zero forcing criterium, $w(z^{-1}) = 1/h(z^{-1})$ as illustrated in Figure 7.4. Hence, no equalization is needed after the decision device: $h'(z^{-1}) = 1$.

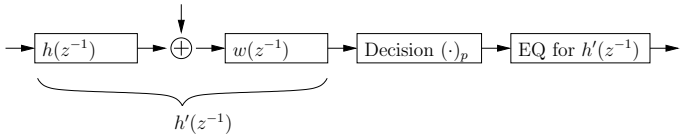


Figure 7.4: Figure indicating three equalization scenarios in the AWGN case: all equalization performed after decision device ($w(z^{-1}) = 1$), all equalization before decision device ($h'(z^{-1}) = 1$) or a mixed scenario using a WMF ($w(z^{-1}) = \frac{h'(z^{-1})}{h(z^{-1})}$).

On the other hand, such an equalizer removing all the ISI before decisions are taken is not a good option either, since it tends to amplify the noise.

- In fact, in a seminal paper [30], Forney shows that the optimal scenario is a mixture of both: part of the equalization is performed by a *Whitening Matched Filter* (WMF) preceding the decision device. The residual ISI can be cancelled afterwards, for example by a Viterbi algorithm (Maximum Likelihood Sequence detector) (See also [30] and Figure 7.5). Again assuming a zero-forcing optimality criterion, the optimal

$$w(z^{-1}) = \frac{h'(z^{-1})}{h(z^{-1})} \tag{7.35}$$

is obtained with *spectral factorization techniques* [59] [46]. These techniques result in an all-pass

$$h'(z^{-1}) = ch_{min}(z)h^*_{max}(1/z^*). \tag{7.36}$$

In this equation, c is a constant, $h_{min}(z)$ and $h^*_{max}(1/z*)$ is the minimum phase (respectively maximum phase) of $h(z^{-1})$.

Note that, around the same time as Forney, Tomlinson [73] and Harashima [40] independently discover the WMF as the optimal front-end in a precoding context. The optimal (finite-length) precoder $h'(z^{-1})$ is found as the solution of an optimization problem which minimizes the noise power N at the input of the decision device:

$$N(\mathbf{h}') = \int_{-W}^{W} N_0 \frac{\mid H'(f) \mid^2}{\mid H(f) \mid^2} df \,. \tag{7.37}$$

The noise reduction compared to a ZF linear equalizer can then be expressed as $G(\mathbf{h}') = \frac{N(1)}{N(\mathbf{h}')}$. Note that in the absence of a length constraint on $h'(z^{-1})$ (or if $h(z^{-1})$ has a limited number of taps), the exact solution of this optimization problem $h'(z^{-1})$ can be verified to be a whitening matched filter (Eq. 7.36). If a Tomlinson-Harashima precoder (THP) is used, the objective function (Eq. 7.37)

is minimized with the first coefficient of $h'(z^{-1})$ constrained to be 1, which is necessary to realize the recursive feedback filter. The residual ISI should not be cancelled anymore, since it is prevented by precoding (See Figure 7.5).

Later, it was recognized that the WMF with the same constraint $h'[0] = 1$ is also the optimal feed forward filter in a DFE-ZF (See Figure 7.5). Here, the residual ISI is cancelled in a nonlinear fashion.

In our approach however, $h'(z^{-1})$ is designed to meet an integer constraint: i.e. the filter coefficients of $h'(z^{-1}) \in \mathbb{Z}$, while the channel can have complex (non-integer) coefficients. For the RS code, only the concatenation of $h(z^{-1})$ with $w(z^{-1})$ matters, yielding a filter $h'(z^{-1})$ with integer coefficients. The Reed-Solomon code *tolerates* the filter $h'(z^{-1})$ as explained in the previous section. I.e. the output of the filter (with the CP removed) is a Reed-Solomon codeword (up to noise) and can be decoded in a maximum likelihood sense (MLSD) or e.g. by a Berlekamp-Massey algorithm (to reduce the complexity).

In conclusion, the underlying causes of the performance gain of an RS-OFDM system over a typical concatenated RS-coded OFDM system are twofold.

- First, in an RS-OFDM system, the transmitted sequences are codewords of an RS code, with a higher distance and consequently better resilience against noise.
- Secondly, the complex field equalizer is only a partial equalizer, allowing a residual integer channel response. Therefore, it prevenst noise amplification and the Galois field equalizer takes care of the *real* equalization in a joint fashion with the RS decoder.

Therefore, it is also instructive to compare the RS-OFDM system with an RS-coded Single Carrier with Cyclic Prefix (SC-CP) system. If the latter system uses the same mapping as in RS-OFDM, both systems transmit codewords of an RS code, such that only the performance gain due to equalization can be observed (second cause). Let us illustrate this with two examples including simulation results.

Example 7.6 Let us construct a 5-band filter bank which implements $\mathcal{R}[10, 5, 6]$ of Example 7.5 with roots are chosen $\{\alpha^k\}_{k=0\ldots4}$. Following the methods presented in Part I of this thesis, the roots $\alpha^0, \alpha^1, \alpha^2, \alpha^3, \alpha^4$ are distributed among the subband filters $\underline{d}_m(z^{-1})$ as follows:

$$\begin{aligned}
\alpha^0 &\Leftarrow \underline{d}_0(z^{-1}) = 1 + \alpha^5 z^{-1} \\
\alpha^1 &\Leftarrow \underline{d}_1(z^{-1}) = \alpha^1 + \alpha^5 z^{-1} \\
\alpha^2 &\Leftarrow \underline{d}_2(z^{-1}) = \alpha^2 + \alpha^5 z^{-1} \\
\alpha^3 &\Leftarrow \underline{d}_3(z^{-1}) = \alpha^3 + \alpha^5 z^{-1}
\end{aligned}$$

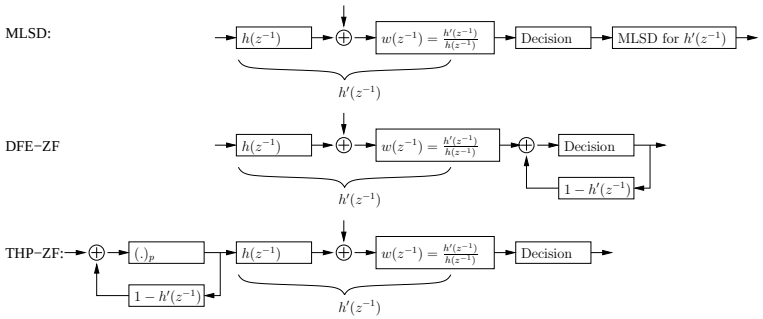


Figure 7.5: A WMF $w(z^{-1})$ is a generic optimal front-end for many equalization techniques, e.g. Maximum Likelihood Sequence Detector (MLSD), Decision Feedback equalization (DFE-ZF) and Tomlinson-Harashima precoding (THP-ZF).

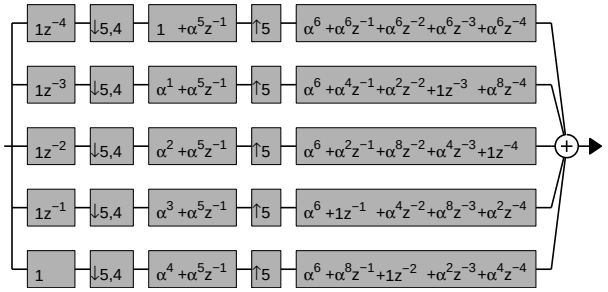


Figure 7.6: Critically subsampled filter bank with 3 bands implementing $\mathcal{R}[10, 5, 6]$ (Example 7.6).

$$\alpha^4 \quad \Leftarrow \quad \underline{d}_4(z^{-1}) = \alpha^4 + \alpha^5 z^{-1}$$

The filter bank so obtained is shown in Figure 7.6. In the more general case (where k and M can be coprime), some subband filters have $\lfloor \frac{k}{M} \rfloor$ taps while others have $\lceil \frac{k}{M} \rceil$. See Chapter 4 for further details.

Let us take a fixed minimum phase channel $h(z^{-1}) = 1.5 + 1.9z^{-1} + 0.7z^{-2}$ (cyclic prefix length 2). The function $G(\mathbf{h}') = G(h'[0], h'[1], h'[2])$ is depicted in Figure 7.7. In our approach, the integer solution of this optimization problem (excluding $h'(z^{-1}) = 0$) is found to be $h'(z^{-1}) = z^{-1} + z^{-2}$. The noise reduction compared to a traditional linear ZF-equalizer $G(0, 1, 1) = 5.64$ dB. Note that when no integer constraint is imposed, but e.g. in the case of a THP-ZF, $h'(z^{-1})$ should be monic, the optimal $h'(z^{-1}) = ch(z^{-1})$ with the constant $c = 1/1.5$. Hence, the noise reduction is calculated to be $G(\mathbf{h}') = 7$ dB compared to the linear zero forcing equalizer ($w(z^{-1}) = 1$).

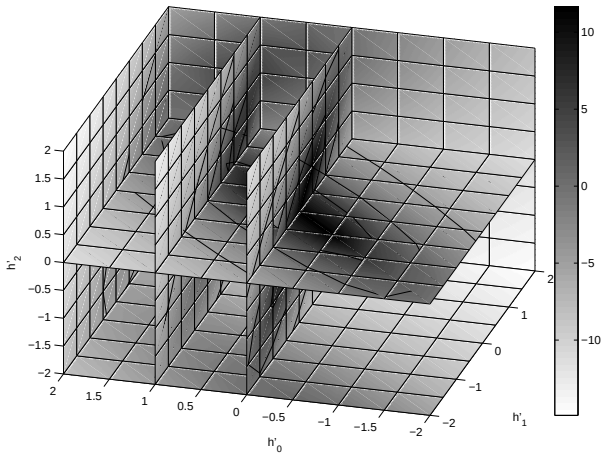


Figure 7.7: The gain $G(\mathbf{h}') = G(h'[0], h'[1], h'[2])$ over a linear ZF equalization as a function of $\mathbf{h}'$ when $h(z^{-1}) = 1.5 + 1.9z^{-1} + 0.7z^{-2}$. The solution of the optimization problem in Eq. 7.37 with an integer constraint is $h'(z^{-1}) = z^{-1} + z^{-2}$ ($G(\mathbf{h}') = 5.64$ dB), with the THP constraint $h'(z^{-1}) = 1 + 1.27z^{-1} + 0.47z^{-2}$ ($G(\mathbf{h}') = 7$ dB)

In Figure 7.8, simulation results are shown. The RS-OFDM based on $\mathcal{R}[10, 5, 6]$ scheme performs approximately 3 dB better than the classical schemes which consist of an OFDM or Single Carrier with Cyclic Prefix (SC-CP) modulator concatenated with a $\mathcal{R}[10, 5, 6]$ code. The Berlekamp-Massey decoder is used to decode $\mathcal{R}[10, 5, 6]$. Note that the gain of 5.64 dB significantly overestimates the actual performance gap between the RS-OFDM and SC-CP system. This is due to the fact that a cyclic prefix of length 2 is quite long compared to the block length $M = 5$. In the next example, a more realistic case with a small cyclic prefix overhead shows better predictions of the actual gain. Secondly, note that in the OFDM case, the transmitted time-domain symbols (and the AWGN) are complex valued, favoring the OFDM case. Also in the next example, all constellations are complex valued such that this degree of freedom is exploited in all 4 cases. △

Example 7.7 As a second example, $\mathbb{F}_{11^2}$ is used (primitive polynomial $\mathscr{P}(x) = 7 + x + x^2$) such that larger constellations and Reed-Solomon codes can be used which are more realistic. More specifically, a rate 3/4 RS code $\mathcal{R}[120, 90, 31]$ in $\mathbb{F}_{11^2}$ is chosen, with roots $\alpha^0..\alpha^{29}$, leading to a 121-QAM constellation. The OFDM block size $M = 15$. Simulation results using $\mathcal{R}[120, 90, 31]$ can be found in Figure 7.9. Again, a minimum phase 4-tap channel $h(z^{-1}) = 1 + (1.29 + 0.80j)z^{-1} + (0.11 + 1.13j)z^{-2} + (-0.47 + 0.43j)z^{-3}$ is chosen. The integer solution of the optimization problem (Eq. 7.37) is found to be $h'(z^{-1}) =$

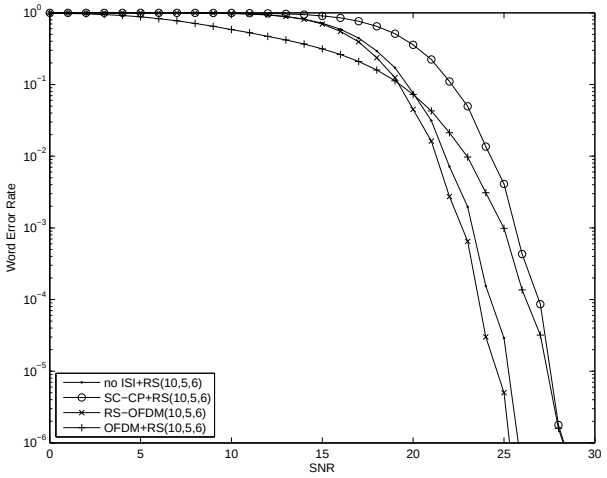


Figure 7.8: Word error rate versus SNR simulation results of RS-OFDM based on $\mathcal{R}\,[10, 5, 6]$ as discussed in Example 7.6. For comparison, a classical OFDM, SC-CP and a flat fading PAM modulated scheme with an outer $\mathcal{R}\,[10, 5, 6]$ are also shown.

$1 + (1 + j)z^{-1} + (-1 + j)z^{-2} - z^{-3}$, with $G(\mathbf{h}') = 10.9\ dB$. In reality, the gain compared to a SC-CP transmission is approximately 10 dB. This gain is only due to the equalization structure in RS-OFDM, which is mainly performed by the RS decoder. There is an additional gain of 4 dB compared to an OFDM based system, due to the distance properties of the transmitted sequences (RS codewords in the RS-OFDM and SC-CP case). △

7.4.2 Block ZF and MMSE Equalizer Design

In the case of a non-minimum phase channel (e.g. a Rayleigh fading channel), the serial ZF equalizer $w(z^{-1})$ does not exist, and can only be approximated with an FIR filter [46]. Nevertheless, the *complex field block equalizer* $\tilde{\mathbf{W}}$ can be designed using an ZF or MMSE criterium. Block equalizers operate in a block-wise fashion on each $M \times 1$ OFDM block. For example, in the ZF case, this equalizer must be designed such that

$$\tilde{\mathbf{W}}\tilde{\mathbf{H}} = \tilde{\mathbf{H}}' \tag{7.38}$$

with $\tilde{\mathbf{H}}'$ and (consequently) $\tilde{\mathbf{W}}$ circulant matrices. Moreover, the complex field equalizer $\tilde{\mathbf{W}}$ should be computed such that $\tilde{\mathbf{H}}'$ is an integer matrix, immediately solving problems related to the integer channel assumption. The RS code allows the residual ISI to be cancelled after decoding, by the *GF equalizer* $\underline{\tilde{\mathbf{E}}}$ now based on $\underline{\tilde{\mathbf{H}}'}$ instead of $\underline{\tilde{\mathbf{H}}}$. Note that $\tilde{\mathbf{W}}$ can be implemented in the frequency domain,

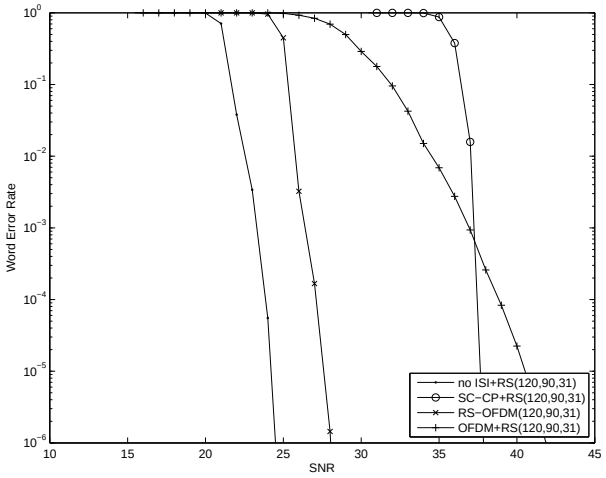


Figure 7.9: Word error rate versus SNR simulation results of RS-OFDM as discussed in Example 7.7. For comparison, a classical OFDM, SC-CP and a flat fading PAM modulated scheme with an outer $\mathcal{R}\,[120, 90, 31]$ are also shown.

just as the FEQ in a traditional OFDM scheme.

$$\underline{\tilde{\mathbf{E}}} = \underline{\mathbf{F}}_M \underline{\tilde{\mathbf{H}}'} \mathbf{F}_M^{-1} \,. \tag{7.39}$$

The rest of this section is devoted to the computation of the complex field equalizer $\tilde{\mathbf{W}}$, based on an MMSE criterion. According to this criterion, the following optimization problem needs to be solved:

$$\min_{\tilde{\mathbf{W}}, \tilde{\mathbf{H}}'} J = \min_{\tilde{\mathbf{W}}, \tilde{\mathbf{H}}'} \mathcal{E}\mathbf{e}^H\mathbf{e} \tag{7.40}$$

subject to $\tilde{\mathbf{H}}'$ integer and $\tilde{\mathbf{W}}, \tilde{\mathbf{H}}'$ circulant. In the latter equation, $\mathcal{E}$ is the expectation operator and the error

$$\mathbf{e} = \mathbf{w} - \hat{\mathbf{w}}. \tag{7.41}$$

The variables $\mathbf{w}$ and $\hat{\mathbf{w}}$ are depicted in Figure 7.10. These are $M \times 1$ vectors corresponding to an OFDM block. This optimization should be solved with an integer constraint on $\tilde{\mathbf{H}}'$, excluding the trivial solutions $\tilde{\mathbf{W}} = \tilde{\mathbf{H}}' = \mathbf{O}$. Apart from the integer constraint, this problem is a standard linear MMSE estimation problem [61]. Since an OFDM transmitted symbol and the noise are both zero-mean random variables, their correlation matrices can be written as

$$\mathbf{R}_\mathbf{n} \quad = \quad \mathcal{E}\mathbf{n}\mathbf{n}^H = \sigma_n^2 \mathbf{I}_M \tag{7.42}$$

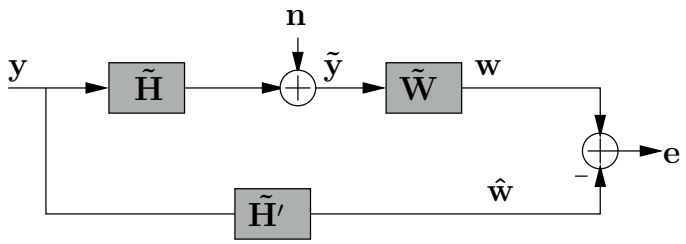


Figure 7.10: Block diagram illustrating the MMSE cost function for finding $\tilde{\mathbf{W}}$.

$$\mathbf{R}_{\mathbf{y}} = \mathcal{E}\mathbf{y}\mathbf{y}^H = \sigma_y^2\mathbf{I}_M = \sigma_u^2\mathbf{I}_M\,. \tag{7.43}$$

Concerning the latter correlation matrix, note that *successive OFDM blocks* $\mathbf{y}$ are correlated (due to the RS subband codes), but there is no correlation *between* the symbols inside an OFDM block. Indeed, any random OFDM block $\mathbf{y}$ can be obtained by properly choosing the corresponding data block $\mathbf{u}$ of length M (See Eq. 7.30). Hence, the codeword symbols are uniformly distributed with energy $\sigma_y^2 = \sigma_u^2$. With $\tilde{\mathbf{y}} = \tilde{\mathbf{H}}\mathbf{y} + \mathbf{n}$, the correlation matrix

$$\mathbf{R}_{\tilde{\mathbf{y}}} = \mathcal{E}\tilde{\mathbf{y}}\tilde{\mathbf{y}}^H \tag{7.44}$$

and the cross correlation matrix

$$\mathbf{R}_{\hat{\mathbf{w}}\tilde{\mathbf{y}}} = \mathcal{E}\hat{\mathbf{w}}\tilde{\mathbf{y}}^H\,. \tag{7.45}$$

The linear least mean squares estimator for $\tilde{\mathbf{W}}$ assuming $\tilde{\mathbf{H}}'$ is given can be written as [61, Section 2.1]

$$\tilde{\mathbf{W}} = \mathbf{R}_{\hat{\mathbf{w}}\tilde{\mathbf{y}}}\mathbf{R}_{\tilde{\mathbf{y}}}^{-1} = \tilde{\mathbf{H}}'\mathbf{R}_{\mathbf{y}\tilde{\mathbf{y}}}\mathbf{R}_{\tilde{\mathbf{y}}}^{-1}\,. \tag{7.46}$$

With

$$\mathbf{R}_{\hat{\mathbf{w}}} = \mathcal{E}\hat{\mathbf{w}}\hat{\mathbf{w}}^H = \tilde{\mathbf{H}}'\mathbf{R}_y\tilde{\mathbf{H}}'^H\,, \tag{7.47}$$

the error covariance matrix becomes

$$\begin{aligned}
\mathbf{R}_e &= \mathcal{E}\mathbf{e}\mathbf{e}^H && (7.48)\\
&= \mathcal{E}(\mathbf{w}-\hat{\mathbf{w}})(\mathbf{w}-\hat{\mathbf{w}})^H && (7.49)\\
&= \mathcal{E}(\mathbf{W}\tilde{\mathbf{y}}-\hat{\mathbf{w}})(\mathbf{W}\tilde{\mathbf{y}}-\hat{\mathbf{w}})^H && (7.50)\\
&= \mathbf{R}_{\hat{\mathbf{w}}} - \mathbf{R}_{\hat{\mathbf{w}}\tilde{\mathbf{y}}}\mathbf{R}_{\tilde{\mathbf{y}}}^{-1}\mathbf{R}_{\hat{\mathbf{w}}\tilde{\mathbf{y}}}^H && (7.51)\\
&= \tilde{\mathbf{H}}'\underbrace{\left(\mathbf{R}_{\mathbf{y}} - \mathbf{R}_{\mathbf{y}\tilde{\mathbf{y}}}\mathbf{R}_{\tilde{\mathbf{y}}}^{-1}\mathbf{R}_{\mathbf{y}\tilde{\mathbf{y}}}^H\right)}_{\mathbf{R}_s}\tilde{\mathbf{H}}'^H\,. && (7.52)
\end{aligned}$$

The matrix $\mathbf{R}_s$ is often referred to as the Schur complement of $\mathbf{R}_e$ with respect to $\mathbf{R}_{\tilde{\mathbf{y}}}$.

The mean square error $\mathcal{E}\mathbf{e}^H\mathbf{e}$ is the trace of the covariance matrix [61, Section 2.1]

$$\min_{\tilde{\mathbf{H}}'} J = \min_{\tilde{\mathbf{H}}'} \mathrm{Tr}(\mathbf{R}_e), \tag{7.53}$$

subject to $\tilde{\mathbf{H}}' \neq \mathbf{0}$, circulant and integer. Assuming positive definiteness of $\mathbf{R}_e$, $\mathbf{R}_s$ will be positive definite too. Consequently, the resulting optimization problem is a mixed integer quadratic programming problem. The global optimum $J_{\min}$ of this optimization problem coincides with the trivial zero solution $\tilde{\mathbf{H}}' = \mathbf{O}$, which is excluded. Since finding an efficient algorithm to solve this problem is out of the scope of this thesis, an exhaustive search starting from the origin is carried out.

Example 7.8 If $h(z^{-1}) = 2.1412 - 0.4400z^{-1} + 4.6143z^{-2}$ and $\sigma_n = 0.3742$ (SNR=20 dB), the solution of the aforementioned optimization problem is

$$\tilde{\mathbf{W}} = \begin{bmatrix} 0.1486 & 0.0097 & 0.1486 & 0.0097 \\ 0.0097 & 0.1486 & 0.0097 & 0.1486 \\ 0.1486 & 0.0097 & 0.1486 & 0.0097 \\ 0.0097 & 0.1486 & 0.0097 & 0.1486 \end{bmatrix} \tag{7.54}$$

$$\tilde{\mathbf{H}}' = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} \tag{7.55}$$

The corresponding minimum mean square error $J_{\min} = 0.0249$. △

The rest of this section compares the performance of RS-OFDM vs. traditional RS coded OFDM and SC-CP. The latter two schemes can be explained in terms of Figure 7.1.

In the case of an OFDM scheme, an RS code $\mathcal{R}[\nu, \kappa]$ is concatenated with a complex field OFDM modulator as described in Section 7.2. At the receiver, only a complex field equalizer $\tilde{\mathbf{E}}^{-1}$ is needed, with the diagonal matrix $\tilde{\mathbf{E}}$ defined corresponding to Eq. 7.5. Posterior to this equalization, decisions are taken and the resulting codeword in the GF is decoded with a Berlekamp-Massey RS decoder.

In the SC-CP case, the DFT operation at the transmitter is shifted to the receiver such that $C_m(z^{-1}) = z^{-m}$. The complex field equalizer $\tilde{\mathbf{W}}$ is a circulant matrix, and can consequently be implemented in the frequency domain:

$$\tilde{\mathbf{W}} = \mathbf{F}_M^{-1}\tilde{\mathbf{E}}^{-1}\mathbf{F}_M. \tag{7.56}$$

Again, the core of the SC-CP system is surrounded with an RS code $\mathcal{R}[\nu, \kappa]$ with a Berlekamp-Massey decoder.

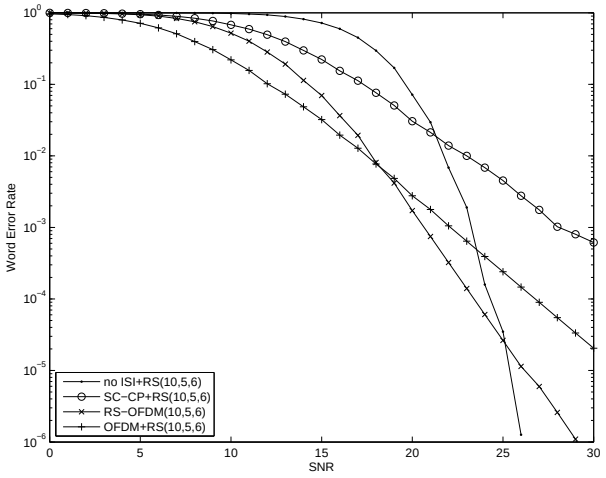


Figure 7.11: Comparison of a concatenated scheme of $\mathcal{R}\,[10, 5]$ followed by an OFDM modulator and the RS-OFDM scheme based on $\mathcal{R}\,[10, 5]$. The RS-OFDM scheme has an overall Hamming distance of 31 compared to 3 for the concatenated scheme.

In Figure 7.11, simulation results are shown. The RS-OFDM scheme based on the RS code $\mathcal{R}\,[10, 5, 6]$ (as introduced in Example 7.5) using an 11-PAM constellation performs approximately 8 dB better than a classical (complex field) OFDM scheme concatenated with the same $\mathcal{R}\,[10, 5]$ code. Likewise, performance results of an RS coded SC-CP system as well as an RS coded system with flat fading are shown (both using an 11-PAM constellation). A 3-tap Rayleigh fading channel is used in all simulations, resulting in a 50% cyclic prefix overhead. In the next example, a more realistic case with a longer RS code and a small cyclic prefix overhead is presented. As a second remark, note that in the OFDM case, the transmitted symbols (and the AWGN) are complex valued, thereby exploiting an extra degree of freedom in favor of the OFDM case. Again, we refer to the next example where all constellations are complex valued.

Simulation results using the the RS code $\mathcal{R}\,[120, 90]$ are shown in Figure 7.12. The constellation is a square 121-QAM constellation. This time, a Rayleigh fading 4-tap channel $h(z^{-1})$ is chosen, with a cyclic prefix length 3. The gain of RS-OFDM over OFDM/SC-CP amounts to approximately 2.5 dB and increases with decreasing word error rate.

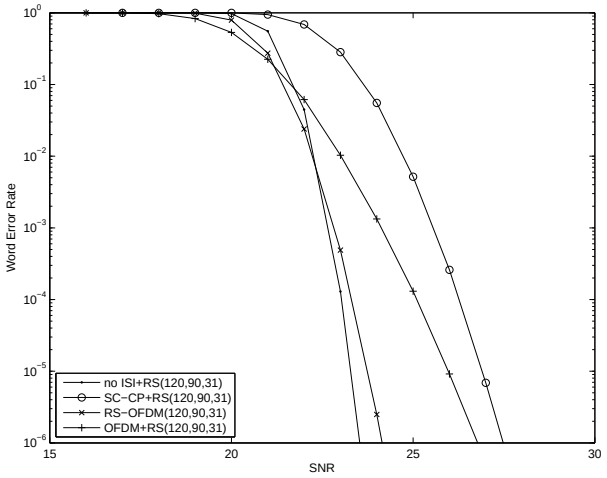


Figure 7.12: Comparison of a concatenated scheme of $\mathcal{R}\,[120, 90]$ followed by an OFDM modulator and the RS-OFDM scheme based on $\mathcal{R}\,[120, 90]$. The RS-OFDM scheme has an overall Hamming distance of 6 compared to 2 for the concatenated scheme.

7.5 Further Research: OFDMA

This section describes a possible future research direction. In case, we present merely a proof of concept of how to extend the RS-OFDM system towards an uplink CDMA system (co-located receiver). It is shown that the spreading codes and the error correcting codes of each user can be jointly designed, such that the overall code is a Reed-Solomon code. Therefore, our scheme is referred to as RS-OFDMA. At the receiver side, a (possibly soft) RS-decoder is applied to decode this overall RS code. Doing so, an attempt is made to develop a coded OFDMA system which is robust to channel fades. It is then often possible to recover the correct dataword of a user experiencing a deep channel fade by decoding the overall RS code, which has an error correcting capability far beyond the capability of an error correcting code on a per user basis.

In fact, an (RS-)OFDMA system is very similar to an (RS-)OFDM system. A closer look at the analysis bank of the RS-OFDM system – which performs basically a serial to parallel conversion – unveils that the frequency bands are separately encoded. This observation makes it possible to extend the RS-OFDM system to RS-OFDMA, with one band allocated per user. A major difference between OFDM and OFDMA is that the different users in an OFDMA system experience a different channel. This is illustrated in Figure 7.13. Hence, the summation of the different user signals is delayed, and occurs in the channel (instead of the transmitter). From a matrix point of view, the matrix $\mathbf{C}$

defining the synthesis bank becomes

$$\mathbf{C} = \left[\begin{array}{cccc} c_0[0] & & & \\ & c_1[0] & & \\ & & \ddots & \\ & & & c_{M-1}[0] \\ \hline c_0[1] & & & \\ & c_1[1] & & \\ & & \ddots & \\ & & & c_{M-1}[1] \\ \hline \vdots & \vdots & \vdots & \vdots \\ \hline c_0[M-1] & & & \\ & c_1[M-1] & & \\ & & \ddots & \\ & & & c_{M-1}[M-1] \end{array}\right]$$

with the user (spreading) codes defined as $c_m(z^{-1}) = \sum_{n=0}^{M-1} c_m[n]\, z^{-n}$ with $c_m[n] = W_M^{-mn}$, W_M^{-mn} an M-th root of unity. If the channel of the m-th user is defined as $h_m(z^{-1}) = \sum_{n=0}^{M-1} h_m(n) z^{-n}$, the channel matrix (cyclic prefix included) becomes

$$\mathbf{H} = \left[\begin{array}{cccc} \mathbf{H}' & \mathbf{R}_M\mathbf{H}' & \mathbf{R}_M^2\mathbf{H}' & \ldots \mathbf{R}_M^{M-1}\mathbf{H}' \end{array}\right],$$

with $\mathbf{R}_M$ representing the rotation matrix as defined in Eq. 4.9 and

$$\mathbf{H}' = \left[\begin{array}{cccc} h_0[0] & h_1[0] & \ldots & h_{M-1}[0] \\ h_0[1] & h_1[1] & \ldots & h_{M-1}[1] \\ \vdots & \vdots & \vdots & \vdots \\ h_0[M-1] & h_1[M-1] & \ldots & h_{M-1}[M-1] \end{array}\right].$$

Note that we assume that the cyclic prefix is longer than the highest channel order L (over all users). To reduce the overhead of the cyclic prefix, note that the OFDM block length M is typically much larger than L. Some algebraic manipulations show that the channel matrix $\mathbf{H}$ is diagonalized by the DFT operation at the receiver:

$$\mathbf{E} = \mathbf{F}_M\mathbf{H}\mathbf{C}\,. \tag{7.57}$$

This DFT operation leads to a convenient way to separate and equalize the different users signals. Moreover, in an RS-OFDMA system, the different user (spreading) codes cooperate with the error correcting codes (per user) to obtain one large RS code. The importance of jointly designed error correcting codes and spreading codes is recognized in [63], but rarely studied.

However, there is an open problem that has still to be solved. It turns out that the factorization in Eq. 7.57 holds only possible for *integer* channel coefficients

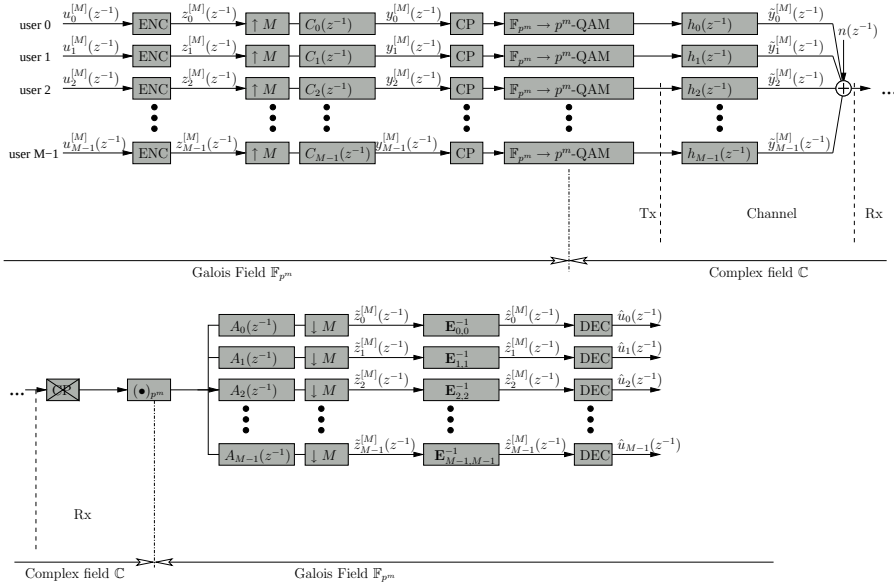
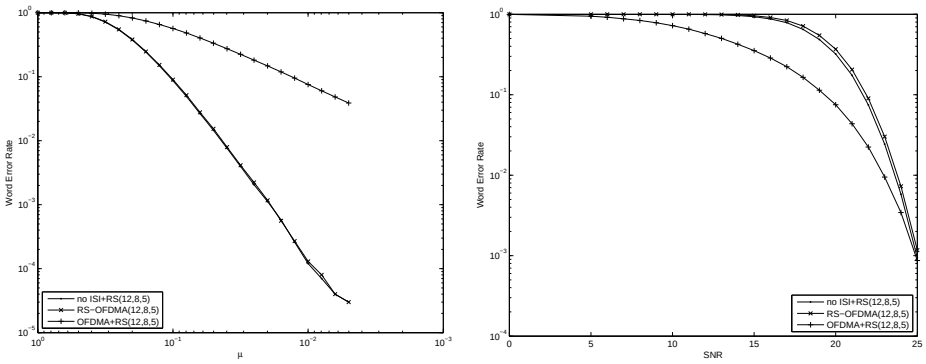


Figure 7.13: Transmultiplexer structure of a baseband discrete-time model of a coded OFDMA system.

since a Galois field counterpart of Eq. 7.57 does not exist for real number valued channel coefficients. Rounding the channel coefficients causes large quantization errors, and is therefore not an option. In conclusion, further research is required to apply the Reed-Solomon codes in a multiuser CDMA scenario. In this context, BCH codes might offer an alternative since they allow the channel to be quantized on a much finer scale than the code alphabet. In the following examples, a comparison between two RS-OFDMA systems is presented to show the potential of this scheme compared to conventional OFDMA schemes with an outer Reed-Solomon code per user.

Example 7.9 This example includes simulation results for the RS-OFDMA system based on the $\mathcal{R}[12, 8, 5]$ code. In the impulse noise case, Figure 7.14(a) shows the word error rate versus the probability of error. The curve corresponding to the RS-OFDMA system has a 3 times steeper slope compared to the RS coded OFDMA system, in which each user has a separate $\mathcal{R}[3, 2, 2]$ code and a complex field DFT spreading code. Moreover, the RS-OFDMA graph coincides with the ISI-free RS coded OFDMA system. Different observations are made in AWGN case (Figure 7.14(b)). The RS-OFDMA performs worse than the classical RS coded (complex-field) OFDMA scheme, unless in the high SNR region where their performance is more or less the same. Two major reasons can be identified for this performance loss: first, the complex field OFDMA transmits constellations in the complex field, exploiting an extra



(a) Case I: Impulse noise (b) Case II: AWGN

Figure 7.14: Simulation results comparing the 4-user RS-OFDMA based on $\mathcal{R}[12, 8, 5]$ system with a conventional (complex field) OFDMA system where each user is coded with a shortened RS(3,2,2) code. The noise is gated impulse noise with SNR=-10 dB.

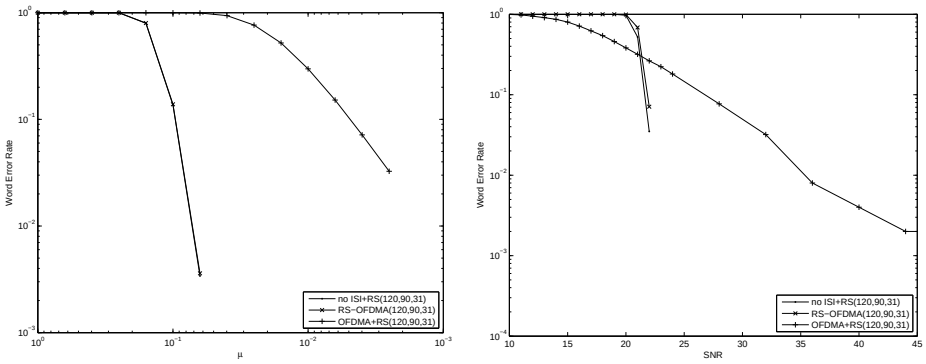
degree of freedom. Secondly, some noise (resulting in symbol errors) is removed by the equalizer which scales the received symbols and thereby the noise. In the Galois field, this scaling is absent; a decision is taken first. This is also the reason why the two graphs approach each other in high SNR regime. However, as illustrated in the next example, if the length of the RS code is chosen in a more realistic way, the coding gain of the RS-OFDMA scheme dominates, resulting in a positive overall performance. △

Example 7.10 The second example is based on $\mathcal{R}[120, 90, 31]$ in $\mathbb{F}_{11^2}$ as defined in Example 7.7 ($M = 15$). Comparison results for the impulse noise case are found in Figure 7.15(a). In this case, the RS-OFDMA system outperforms the RS coded classical complex field OFDMA system. Also in the AWGN case (Figure 7.14(b)), the difference between the slopes is clear.

△

7.6 Conclusion

In this chapter, a second application of filter bank decompositions of RS codes has been presented. The starting point is an OFDM scheme in which the circulant integer channel matrix is decomposed using DFT matrices in a Galois field instead of the complex field, called GF-OFDM. In order to be compatible with the finite field operations, the channel is assumed to be quantized and a modulo operation must be added at the receiver. Using a filter bank representation



(a) Case I: Impulse noise (b) Case II: AWGN

Figure 7.15: Simulation results comparing the 15-user RS-OFDMA system with a conventional (complex field) OFDMA system where each user is coded with a shortened RS(8,6,3) code of the same rate $\frac{3}{4}$. The noise is gated impulse noise with SNR=-10 dB.

of an RS code, it is explained how this GF-OFDM scheme can be seamlessly merged with an RS code. The overall scheme, referred to as RS-OFDM, shows an RS code which matches the OFDM-modulator. The performance gain as a result of this is twofold. First, the optimal Hamming distance of the RS code is preserved by the OFDM modulator. Secondly, the equalization can now be performed in a joint fashion by the RS decoder, avoiding noise amplification. Simulation results confirm a significant performance gain (in some cases up to 10 dB) over classical RS coded OFDM and SC-CP schemes. In addition to the performance gain, an RS-OFDM scheme transmits a PAM or QAM constellation, with a reduced PAPR (0 dB). Moreover, the complexity can be reduced since the whole RS-OFDM system can be seen as one large RS code, such that low complexity RS decoders can be applied in practice.

In a more realistic scenario with a non-integer channel, a modified RS-OFDM system is presented in which the equalization is split in two stages: the first stage encompasses a partial equalization in the complex field, avoiding noise amplification. This equalizer also ensures that the residual channel response has integer coefficients. It is calculated using a zero forcing (ZF) or Minimum Mean Square Error (MMSE) criterion. In the second stage, the residual ISI as well as the remaining noise is removed by the Reed-Solomon decoder followed by a Galois field equalizer. The performance of the RS-OFDM system is further evaluated by simulation results showing a significant performance gain of the RS-OFDM system compared to classical coded OFDM/SC-CP schemes.

Chapter 8

Conclusions

In this thesis, the filter bank structure behind Reed-Solomon codes is uncovered. Unlike classical *overlap-add* and *overlap-save* filter banks, a *critically* subsampled filter bank representation of an RS code is obtained. This is remarkable noting that a critically subsampled bank can never perform linear filtering operations. In fact, the periodically time varying behavior resulting from the critical subsampling matches with the cyclic character of the RS code. The synthesis and analysis bank are based on the *Mattson-Solomon polynomial* or DFT in the Galois field and on the fact that it can be used to diagonalize a circulant matrix. Due to the critical subsampling, there is a clear distinction between the subband codes (responsible for the addition of redundancy) and how this redundancy is further combined (synthesis bank). This property is crucial in the applications discussed in the second part of the thesis.

Moreover, using an algebraic approach, we have shown that the roots of the RS code are evenly distributed over the subband codes, which are in turn RS-like codes. This useful property provides further insight into the algebraic structure of the RS code as well as into the corresponding filter bank realization. Two decompositions take a central role in this approach: the FFT and Cook-Toom's algorithm. A key element in this derivation is the exploitation of the cyclic properties of the RS code, in order to find a block generator matrix of an equivalent RS code. The FFT operation transforms the cyclic character modulo the block length into a cyclic behavior modulo a divisor of this block length. This approach allows us to show that the subband codes are generalized non-primitive BCH codes, with properties similar to RS codes.

There is an important link with earlier work on the block circulant structure of RS codes by Solomon and Van Tilborg. Although the latter structure does only exist for a subset of RS codes (with the block length ν and dimension

κ not coprime), the filter bank based techniques presented in this thesis can be generally applied to *any* RS code. Systematic filter banks are also briefly discussed.

In a next step, an extension to the more general class of BCH codes, to which RS codes belong, is presented. BCH codes have a cyclic character too, but the block length ν can now differ from $q-1$ imposing a number of constraints related to the existence of the DFT. Nevertheless, this thesis shows that each BCH code can be implemented as a sum of critically subsampled filter banks.

As a result of this, another important connection with work by Berlekamp and Justesen on 2-D cyclic (concatenated) codes is studied, in which an inner and outer cyclic code are concatenated to obtain one large cyclic code. If a subband code is associated with an outer code while the inner code corresponds to the synthesis bank filters, we show that each (co)set of nonzeros of the overall BCH code is realized by a specific combination of the cosets of inner and outer code. These results make us conclude that filter banks are very powerful tools – if not the most appropriate tools – to describe BCH and RS codes.

The usefulness of these filter banks in practical applications is illustrated in the second part of this thesis.

The first problem looked into is SISO RS decoding. Standard so-called hard RS decoders are unable to take into account additional information carried by an erroneously received codeword symbol, e.g. the correct symbol might be (with a high probability) in the vicinity of the erroneous one. A classical solution is to concatenate the RS code with an inner convolutional code that is easily soft decodable. Unfortunately, the distance of the overall code can hardly be assessed, and is typically much lower than what can be achieved due to the small constraint length of the inner code. At the other hand, it is well known that the concatenation principle paves the way for low complexity SISO decoding, e.g. in LDPC and Turbo codes. However, also for these codes, there is little control regarding the overall code.

A solution to this problem is offered in this thesis, based on the filter bank representation of RS codes. More specifically, the filter bank structure allows us to decompose the RS code into many small component codes, resulting in a concatenated code structure. This type of concatenation is called *coordinated concatenation*, since all component codes now cooperate yielding an overall RS code. Each of these component codes can be SISO decoded at a low complexity, and exchange information to obtain a SISO decoder of the overall RS code in a Gallager-type algorithm. Typical soft decoding gains are in the order of 1.5 dB compared to hard Berlekamp-Massey decoding (for BPSK modulation). The complexity of the algorithm is dominated by the complexity of a SISO decoder of the largest component code. In this context, it is important that the filter bank decomposition can be applied recursively.

A similar kind of problem is encountered in the second application. Due to the frequency selective character of the communication channel, advanced modulators and equalizers have been designed, e.g. OFDM. Typically, the OFDM system is used in concatenation with an outer error correcting code, e.g. an RS code. Although OFDM with bit-loading is known to achieve capacity (i.e. with a finite length, rate 0 theoretical code), the situation is quite different if a finite length code is used. Once again, the distance properties of the overall code must be carefully assessed and optimized.

In this thesis, a solution to this problem is presented. It is explained how an OFDM scheme can seamlessly be merged with a Reed-Solomon code. Using the filter bank representation of RS codes, it is easily seen that the DFT hidden in the synthesis bank can be (re)used by the OFDM modulator. The overall optimal scheme so obtained is referred to as RS-OFDM, and shows an RS code that is matched to the OFDM-modulator. This jointly designed RS-OFDM scheme results in a significant performance gain, often more than 5 dB as shown by simulations. In addition to the performance gain, the PAPR of the RS-OFDM scheme is 0 dB, since a PAM/QAM constellation is transmitted. Moreover, a complexity reduction can be achieved since the RS-OFDM system can be seen as one large RS code, such that low complexity RS decoders can be applied in practice. To overcome differences in finite and complex field arithmetic of the RS code and channel respectively, a modulo must be added at the receiver. Moreover, an additional (complex field) equalizer must be added preceding this modulo operation. Some preliminary results show that RS-OFDM can be extended to a multiuser scenario.

Further research

Concerning the SISO RS decoder, a major source of performance degradation is due to cycles in the Tanner graph. As a first step, EXIT chart analysis can lead to a better understanding of the convergence issues of the SISO RS decoder. Another major breakthrough in this field is to look at Gallager's algorithm from an optimization point of view. Recently, it is shown that Gallager's algorithm aims at minimizing an energy function referred to as the *Bethe free energy*. Hence, more advanced optimization methods can be called upon, that provably descend the *Bethe free energy*, thereby avoiding instabilities as encountered now. Another interesting approach in this context is due to Sudderth, that solves the inference problem exactly by solving a series of modified estimation problems on (cycle free) spanning trees embedded in the original cyclic graph. Also note that other low complexity SISO algorithms can be plugged in to replace the existing optimal SISO decoders of the component codes. As such, the use of the filter bank decomposition is not limited to Gallager type of algorithms.

Apart from increasing the decoding performance, the proposed methods should

also be evaluated for longer RS codes. If the block length of the RS code increases, the order of the Galois field grows correspondingly, resulting in many negligible entries in the reliability matrices. Quantization of these matrices results in a further reduction of the memory requirements of the algorithm, but also (inevitably) in a performance degradation. A further evaluation of the algorithm with quantized reliability matrices is therefore required. In this respect, BCH codes (which show a similar filter bank structure) might offer an interesting alternative, since the block length of a BCH code can be chosen (quasi) independently from the order of the Galois field.

Concerning RS-OFDM, preliminary results indicate a possible extension towards RS-OFDMA, with multiple users. An open problem is this context is the quantization of the channel coefficients. A particular matrix factorization, which is crucial for the RS-OFDMA concept to work, does not hold for real number valued channel coefficients. Further research is required to apply the Reed-Solomon codes in a multiuser CDMA scenario. In this context, the filter bank decomposition of BCH codes might offer an alternative since they allow the channel to be quantized on a much finer scale than the code alphabet.

Bibliography

[1] Digital video broadcasting: Framing structure, channel coding, and modulation for digital terrestrial television. *ETSI, EN 300-74*, Aug. 1997.

[2] Radio broadcasting systems; DAB to mobile, portable and fixed. receivers. *ETSI, ETS 300-401*, May 1997.

[3] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv. Optimal decoding of linear codes for minimizing symbol error rate. *IEEE Trans. Inform. Theory*, 20(2):284 –287, March 1974.

[4] S. Benedetto, D. Divsalar, and J. Hagenauer. Guest Editorial Concatenated Coding Techniques And Iterative Decoding: Sailing Toward Channel Capacity. *IEEE J. Select. Areas Commun.*, 16(2):137 – 139, Feb 1998.

[5] E. Berlekamp and J. Justesen. Some long cyclic linear binary codes are not so bad. *IEEE Trans. Inform. Theory*, 20(3):351 – 356, May 1974.

[6] Elwyn R. Berlekamp. *Algebraic Coding Theory.* McGraw-Hill, New York, 1968.

[7] C. Berrou and A. Glavieux. Near optimum error correcting coding and decoding: Turbo-Codes. *IEEE Trans. Commun.*, 44(10):1261 –1271, October 1996.

[8] R.E. Blahut. Transform Techniques for Error-Control Codes. *IBM J. Res. Develop.*, 23:299–315, May 1979.

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

[10] I.M. Boyarinov and G.I. Katsman. Linear Unequal Error Protection Codes. *IEEE Trans. Inform. Theory*, 27(2):168–175, March 1981.

[11] U. Cheng and G.K. Huth. Bounds on the bit error probability of a linear cyclic code over GF(2^l) and its extended code. *IEEE Trans. Inform. Theory*, 34(4):776 – 785, July 1988.

[12] A. Chouly, A. Brajal, and S. Jourdan. Orthogonal Multicarrier Techniques Applied to Direct Sequence Spread Spectrum CDMA Systems. In *Proc. of IEEE Global Telecommunications Conference (GLOBECOM), Houston, USA*, volume 3, pages 1723–1728, Nov 1993.

[13] J.S. Chow, J.C. Tu, and J.M. Cioffi. A discrete multitone transceiver system for HDSL applications. *IEEE J. Select. Areas Commun.*, 9(6):895–908, Aug. 1991.

[14] Sae-Young Chung, T.J. Richardson, and R.L. Urbanke. Analysis of sum-product decoding of Low-Density Parity-Check codes using a Gaussian approximation. *IEEE Trans. Inform. Theory*, 47(2):657 – 670, Feb 2001.

[15] J. Conway, V. Pless, and N. Sloane. Self-dual codes over GF(3) and GF(4) of length not exceeding 16. *IEEE Trans. Inform. Theory*, 25(3):312 – 322, May 1979.

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

[17] V. DaSilva and E.S. Sousa. Performance of Orthogonal CDMA Codes for Quasi-Synchronous Communication Systems. In *Proc. of 2nd IEEE International Conference on Universal Personal Communications (KUPC) 1993, Ottawa, Canada*, volume 2, pages 995–999, Oct. 1993.

[18] J.A. Davis and J. Jedwab. Peak-to-mean power control in OFDM, Golay complementary sequences, and Reed-Muller codes. *IEEE Trans. Inform. Theory*, 45(7):2397 – 2417, Nov 1999.

[19] Changyan Di, D. Proietti, I.E. Telatar, T.J. Richardson, and R.L. Urbanke. Finite-length analysis of Low-Density Parity-Check codes on the binary erasure channel. *IEEE Trans. Inform. Theory*, 48(6):1570 – 1579, June 2002.

[20] S. Dolinar, D. Divsalar, and F. Pollara. Code performance as a function of block size. *The Telecommunications and Mission Operations Progress Report PR 42-133*, January-March 1998:1–23, May 1998.

[21] L.A. Dunning and W.E. Robbins. Optimal encodings of linear block codes for unequal error protection. *Inform. Contr.*, 37:150–177, 1978.

[22] M. El-Khamy and R.J. McEliece. Iterative algebraic soft-decision list decoding of Reed-Solomon codes. *IEEE J. Select. Areas Commun.*, 24(3):481 – 490, March 2006.

[23] R.M. Fano. *Transmission of Information*. Jointly published by the MIT Press and John Wiley and Sons, 1961.

[24] K. Fazel and L. Papke. On the Performance of Convolutionally-Coded CDMA/OFDM for Mobile Communication System. In *Proc. of IEEE PIMRC, Yokohama, Japan*, pages 468–72, Sept 1993.

[25] F. Fekri, S.W. McLaughlin, R.M. Mersereau, and R.W. Schafer. Block error correcting codes using finite-field wavelet transforms. *IEEE Trans. Signal Processing*, 54(3):991 – 1004, March 2006.

[26] F. Fekri, R.M. Mersereau, and R.W. Schafer. Two-band wavelets and filterbanks over finite fields with connections to error control coding. *IEEE Trans. Signal Processing*, 51(12):3143 – 3151, Dec 2003.

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

[28] G.D.Jr. Forney. Codes on graphs: normal realizations. *IEEE Trans. Inform. Theory*, 47(2):520–548, Feb 2001.

[29] G.Jr. Forney. Generalized minimum distance decoding. *IEEE Trans. Inform. Theory*, 12(2):125 – 131, April 1966.

[30] G.Jr. Forney. Maximum-likelihood sequence estimation of digital sequences in the presence of intersymbol interference. *IEEE Trans. Inform. Theory*, 18(3):363 – 378, May 1972.

[31] M.P.C. Fossorier and Shu Lin. Soft-input soft-output decoding of linear block codes based on ordered statistics. In *Proc. of the IEEE Global Telecommunications Conference (Globecom)*, volume 5, pages 2828 – 2833, 8-12 Nov 1998.

[32] R. Gallager. A simple derivation of the coding theorem and some applications. *IEEE Trans. Inform. Theory*, 11:318, Jan 1965.

[33] R. Gallager. *Information Theory and Reliable Communication*. New York, NY: John Wiley and Sons, 1971.

[34] R.G. Gallager. *Low Density Parity Check Codes*. Monograph, M.I.T. Press, 1963.

[35] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. Pattern Anal. Machine Intell.*, 6:721–741, 1984.

[36] K. S. Gilhousen, I. M. Jacobs, R. Padovani, A.J. Viterbi, L.A.Jr. Weaver, and III Wheatley, C.E. On the capacity of a cellular CDMA system. *IEEE Trans. Veh. Technol.*, 40(2):303–312, May 1991.

[37] V. Guruswami and M. Sudan. Improved decoding of Reed-Solomon and algebraic-geometric codes. *IEEE Trans. Info. Theory*, 45:1757–1767, 1999.

[38] J. Hagenauer, E. Offer, and L. Papke. Iterative decoding of binary block and convolutional codes. *IEEE Trans. Inform. Theory*, 42(2):429 – 445, March 1996.

[39] T.R. Halford, V. Ponnampalam, A.J. Grant, and K.M. Chugg. Soft-in soft-out decoding of Reed-Solomon codes based on Vardy and Be'ery's decomposition. *IEEE Trans. Inform. Theory*, 51(12):4363 – 4368, dec 2005.

[40] H. Harashima and H. Miyakawa. Matched transmission technique for channels with Intersymbol Interference. *IEEE Trans. Commun.*, 20:774–780, August 1972.

[41] E. Ising. Beitrag zur Theorie des Ferromagnetismus. *Zeitschr. f. Physik*, 31:253–258, 1925.

[42] Jing Jiang and K.R. Narayanan. Iterative soft decoding of Reed-Solomon codes. *IEEE Communications Letters*, 8(4):244 – 246, April 2004.

[43] A.E. Jones, T.A. Wilkinson, and S.K. Barton. Block coding scheme for reduction of peak to mean envelope power ratio of multicarrier transmission schemes. *Electronics Letters*, 30(25):2098–2099, Dec 1994.

[44] R. Koetter and A. Vardy. Algebraic soft-decision decoding of Reed-Solomon codes. *IEEE Trans. Inform. Theory*, 49(11):2809 – 2825, Nov 2003.

[45] F. Labeau, L. Vandendorpe, and B. Macq. Oversampled filter banks as error correcting codes. In *The 5th International Symposium on Wireless Personal Multimedia Communications*, volume 3, pages 1265 – 1269, 2002.

[46] E.A. Lee and D.G. Messerschmitt. *Digital Communication*. Kluwer Academic Press, Boston, 1988.

[47] T.Jr. Marshall. Coding of Real-Number Sequences for Error Correction: A Digital Signal Processing Problem. *IEEE J. Select. Areas Commun.*, 2(2):381 – 392, Mar 1984.

[48] B. Masnick and J. Wolf. On linear unequal protection codes. *IEEE Trans. Inform. Theory*, 13:600–607, Oct. 1967.

[49] J. L. Massey. Shift-Register Synthesis and BCH Decoding. *IEEE Trans. Inform. Theory*, 15(1):122–127, Jan 1969.

[50] R.J. McEliece, D.J.C. MacKay, and Jung-Fu Cheng. Turbo decoding as an instance of Pearl's Belief Propagation algorithm. *IEEE J. Select. Areas Commun.*, 16(2):140 –152, February 1998.

[51] T. Ojanpera and R. Prasad. An overview of air interface multiple access for IMT-2000/UMTS. *IEEE Commun. Mag.*, 36(9):82–86, Sept. 1998.

[52] K.G. Paterson and V. Tarokh. On the existence and construction of good codes with Low Peak-to-Average Power ratios. *IEEE Trans. Inform. Theory*, 46(6):1974–1987, Sept. 2000.

[53] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1988.

[54] V. Ponnampalam and A. Grant. An efficient SISO algorithm for Reed-Solomon codes. In *Proceedings of the IEEE International Symposium on Information Theory*, page 204, 29 June-4 July 2003.

[55] V. Ponnampalam and B. Vucetic. Soft decision decoding of Reed-Solomon codes. *IEEE Trans. Commun.*, 50(11):1758 – 1768, Nov 2002.

[56] M.R. Portnoff. Time-frequency representation of digital signals and systems based on short-time Fourier analysis. *IEEE Trans. Acoust., Speech, Signal Processing*, 35:356–372, March 1987.

[57] I. Reed and G. Solomon. Polynomial codes over certain finite fields. *SIAM J.*, 8(2):300–304, June 1960.

[58] T.J. Richardson and R.L. Urbanke. The capacity of Low-Density Parity-Check codes under message-passing decoding. *IEEE Trans. Inform. Theory*, 47(2):599 – 618, Feb 2001.

[59] J. Salz. Optimum mean square decision feedback equalization. *Bell Syst. Tech. J.*, 52:1341–1373, Oct. 1973.

[60] H. Sari and G. Karam. Orthogonal Frequency-Division Multiple Access and its application to CATV network. *European Transactions on Telecommunications*, 9:507–516, Nov./Dec. 1998.

[61] A. H. Sayed. *Fundamentals of Adaptive Filtering*. John Wiley & Sons, NY, 2003.

[62] A. Scaglione, G.B. Giannakis, and S. Barbarossa. Redundant filterbank precoders and equalizers. i. unification and optimal designs. *IEEE Trans. Signal Processing*, 47(7):1988 – 2006, July 1999.

[63] C. Schlegel and L. Wei. A simple way to compute the minimum distance in multiuser CDMA systems. *IEEE Trans. Commun.*, 45(5):532 – 535, May 1997.

[64] C. Shannon. A Mathematical Theory of Communication. *The Bell system Technical Journal*, 27:379–423, 623–656, July-October 1948.

[65] C. Shannon, G. Gallager, and E. Berlekamp. Lower bounds to error probability for coding on discrete memoryless channels, Parts I and II. *Information and Control*, 10:65–103, February/ May 1967.

[66] C.E. Shannon. Probability of error for optimal codes in a Gaussian channel. *Bell System Technical Journal*, 38:611–656, May 1959.

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

[68] G. Solomon and H. C. A. van Tilborg. A connection between block and convolutional codes. *SIAM J. Appl. Math*, 37(2):358–369, October 1979.

[69] E.B. Sudderth, M.J. Wainwright, and A.S. Willsky. Embedded trees: estimation of Gaussian Processes on graphs with cycles. *IEEE Trans. Signal Processing*, 52(11):3136 – 3150, Nov. 2004.

[70] R. M. Tanner. A recursive approach to low complexity codes. *IEEE Trans. Inform. Theory*, 27(5):533 – 547, Sep. 1981.

[71] S. ten Brink. Convergence of iterative decoding. *Electronics Letters*, 35(10):806 – 808, May 1999.

[72] L. Thibault and Minh Thien Le. Performance evaluation of COFDM for digital audio broadcasting. I. Parametric study. *IEEE Trans. Broadcast.*, 43(1):64–75, March 1997.

[73] T. Tomlinson. New automatic equalizer employing modulo arithmetic. *Electron. Lett.*, 7:138–139, March 1971.

[74] W. Van Gils. Two topics on linear unequal error protection codes: Bounds on their length and cyclic code classes. *IEEE Trans. Inform. Theory*, 29(6):866–876, nov 1983.

[75] W.J. Van Gils. Linear Unequal Error Protection codes from shorter codes. *IEEE Trans. Inform. Theory*, 30(3):544–546, May 1984.

[76] L. Vandendorpe. Multitone Direct Sequence CDMA System in an Indoor Wireless Environment. In *Proc. of IEEE First Symposium of Communications and Vehicular Technology in the Benelux, Delft, The Netherlands*, pages 4.1–1–4.1.8, Oct. 1993.

[77] A. Vardy and Y. Be'ery. Bit-level soft-decision decoding of Reed-Solomon codes. *IEEE Trans. Commun.*, 39(3):440 –444, March 1991.

[78] M. Vetterli. Running FIR and IIR filtering using multirate filter banks. *IEEE Trans. Acoust., Speech, Signal Processing*, 36(5):730 –738, May 1988.

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

[80] S. Weinstein and P. Ebert. Data Transmission by Frequency-Division Multiplexing Using the Discrete Fourier Transform. *IEEE Trans. Commun.*, 19(5):628–634, Oct. 1971.

[81] N. Wiberg. *Codes and Decoding on General Graphs*. PhD thesis, Linkoping University, Sweden, 1996.

[82] J.M. Wozencraft and B. Reiffen. *Sequential decoding*. MIT Press, Cambridge, MA, 1961.

[83] J.S. Yedidia, W.T. Freeman, and Y. Weiss. Constructing free-energy approximations and generalized belief propagation algorithms. *IEEE Trans. Inform. Theory*, 51(7):2282 – 2312, July 2005.

[84] N. Yee, J-P. Linnartz, and G. Fettweis. Multicarrier CDMA in Indoor Wireless Radio Networks. In *Proc. of IEEE PIMRC, Yokohama, Japan*, pages 109–113, Sept 1993.

[85] W.Y. Zou and Yiyan Wu. COFDM: an overview. *IEEE Trans. Broadcast.*, 41(1), March 1995.

Publications

International Journal Publications

- G. Van Meerbergen, M. Moonen, and H. De Man. Reed-Solomon Codes implementing a Coded OFDM scheme. *submitted to IEEE Transactions on Communications*, 2006.

- G. Van Meerbergen, M. Moonen, and H. De Man. Filterbank Decompositions for (Non)-Systematic Reed-Solomon Codes with Applications to Soft Decoding. *submitted to IEEE Transactions on Signal Processing*, 2005.

- G. Van Meerbergen, M. Moonen, and H. De Man. Critically subsampled filterbanks for SISO Reed-Solomon decoding. *Accepted for publication in IEEE transactions on Signal Processing*, 2005.

- G. Van Meerbergen, M. Vergauwen, M. Pollefeys, and L. Van Gool. A hierarchical symmetric stereo algorithm using dynamic programming. *International journal of computer vision - special issue on stereo and multi-baseline vision, G. Bradski, T. Boult*, 47(1-3):275 – 285, May 2002.

International Conference Publications

- G. Van Meerbergen, M. Moonen, and H. De Man. Reed-Solomon Codes implementing a Coded OFDM scheme for Rayleigh fading channels. In *Accepted for publication in the proceedings of the IEEE Global Telecommunications Conference (Globecom 2006), San Francisco, USA*, Nov. 27-Dec. 1 2006.

- G. Van Meerbergen, M. Moonen, and H. De Man. RS-OFDM: Using Reed-Solomon codes as coded OFDM modulator. In *Proc. of the IEEE Benelux Signal Processing symposium*, volume 0, March 28-29 2006.

- G. Van Meerbergen, M. Moonen, and H. De Man. Combining Reed-Solomon codes and OFDM for impulse noise mitigation: RS-OFDM. In *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toulouse, France*, volume 4, pages 657–660, May 15-19 2006.

- G. Van Meerbergen, M. Moonen, and H. De Man. Filterbank Decompositions for BCH-codes with Applications to Soft Decoding and Code Division Multiple Access systems. In *Proc. of the IEEE International Symposium on Information Theory, Adelaide, Australia*, pages 2389 – 2393, Sept. 4-9 2005.

- G. Van Meerbergen, M. Moonen, and H. De Man. Soft-In Soft-Out Reed-Solomon decoding using Critically Subsampled Filterbanks. In *Proc. of the IEEE Information Theory Workshop on Coding and Complexity (ITW), Rotorua, New-Zealand*, pages 126 – 131, Aug. 29 - Sept. 1 2005.

- G. Van Meerbergen, M. Moonen, and H. De Man. IIR Critically Subsampled Filterbanks implementing Systematic Reed-Solomon codes. In *Proc. of the IEEE International Conference on Communications (ICC), Seoul, Korea*, volume 1, pages 698 – 703, May 16-20 2005.

- G. Van Meerbergen, M. Moonen, and H. De Man. Critically subsampled filterbanks implementing Reed-Solomon codes: An algebraic point of view. In *Proc. of the IEEE Benelux Signal Processing symposium*, volume 0, pages 115–119, April 19-20 2005.

- G. Van Meerbergen, M. Moonen, and H. De Man. Critically subsampled filterbanks implementing Reed-Solomon codes: An algebraic point of view. In *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2005), Philadelphia, USA*, volume 4, pages 581–584, March 19-23 2005.

- G. Van Meerbergen, M. Moonen, and H. De Man. Turbo-like Soft-Decision Decoding of Reed-Solomon Codes. In *Proc. of the IEEE Global Telecommunications Conference (Globecom 2004), Dallas, USA*, volume 1, pages 199 – 203, Nov. 29-Dec. 3 2004.

- G. Van Meerbergen, M. Moonen, and H. De Man. Turbo-like Soft-Decision Decoding of Reed-Solomon Codes. In *Proc. of the Symposium on Information Theory in the Benelux, Kerkrade, The Netherlands*, pages 9–16, June 2004.

- G. Van Meerbergen, M. Moonen, and H. De Man. Critically subsampled filterbanks implementing Reed-Solomon codes. In *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Montreal, Canada*, volume 2, pages 989–992, May 17-21 2004.

- G. Van Meerbergen, M. Moonen, and H. De Man. Critically sampled filterbank representation of Reed-Solomon codes. In *Proc. of the IEEE Benelux Signal Processing symposium (SPS), Hilvarenbeek, The Netherlands*, pages 149–153, April 2004.

- G. Van Meerbergen, M. Moonen, and H. De Man. Soft-Decision Decoding of Reed-Solomon codes based on Filterbanks. In *Proc. of the Conference on Information Sciences and Systems (CISS), Princeton, USA*, pages 287–292, March 17-19 2004.

- G. Van Meerbergen, M. Vergauwen, M. Pollefeys, and L. Van Gool. A hierarchical stereo algorithm using dynamic programming. In *Proc. of the IEEE Workshop on Stereo and Multi-Baseline Vision (SMBV), Hawaii, USA*, pages 166 – 174, 9-10 Dec. 2001.

Curriculum Vitae

Geert Van Meerbergen was born in Lier, Belgium, in 1978. In 2001 he received the electrical engineering degree (summa cum laude) from the Katholieke Universiteit Leuven, Belgium. Since 2001 he has been pursuing a Ph.D. at the Electrical Engineering Department (ESAT) of the Katholieke Universiteit Leuven, under supervision of Prof. dr. ir. Marc Moonen. His research interests are in the area of digital signal processing techniques for communications and coding.

From 2001 until 2005 he was funded by the F.W.O. (Fund for Scientific Research - Flanders). In 2001 he received the Krypton award for his MSc thesis and in 2004 he received the Gauss award (Best Student Paper Award) at the 25-th Symposium on Information Theory in the Benelux. He was also the recipient of an IEEE Travel Grant in 2005 for the IEEE International Conference on Communications, Seoul.