



MASTER'S THESIS

SAMPLING OF SPARSE ANALOG SIGNALS

performed at
Signal Processing and Speech Communications Laboratory
Graz University of Technology

by
Nikolaus HAMMLER, BSc

Supervisor:
Dipl.-Ing. Dr. Assoc.Prof. Klaus WITRISAL

Advisor:
Dipl.-Ing. Dr. Christian VOGEL

Graz, January 2011

Statutory Declaration

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

date


(signature)

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommene Stellen als solche kenntlich gemacht habe.

Graz, am

(Unterschrift)

Signature Value	JrXiUWzMnoc9EiOw8xDRQcD+w1E0BpkYQ0qKF9CE75xBMF6uW0/2HpvY17/R2GEO	
	Signatory	T=B.Sc., serialNumber=985939835122, givenName=Nikolaus, SN=Hammler, CN=Nikolaus Hammler, C=AT
	Date/Time-UTC	2011-01-18T21:30:21Z
	Issuer-Certificate	CN=a-sign-Premium-Sig-02, OU=a-sign-Premium-Sig-02, O=A-Trust Ges. f. Sicherheitssysteme im elektr. Datenverkehr GmbH, C=AT
	Serial-No.	315810
	Method	urn:pdfsigfilter:bka.gv.at:binaer:v1.1.0
	Parameter	etsi-bka-1.0@1295386222-199262694@20179-11310-0-4984-16680
Verification	Verification service: https://www.signaturpruefung.gv.at	

Abstract

Conventional sampling systems are based on the Whittaker-Kotelnikov-Shannon (WKS) sampling theorem which states signals should be sampled at twice the highest frequency component. In fact, this is a worst-case assumption. It applies to (bandlimited) white noise as well as to signals which contain only a small amount of information.

Many signals can be described with considerably less parameters than predicted by the WKS theorem. This is achieved by describing the data with an appropriate representation (for example, using a transform like the Fourier transform) in which the signal is *sparse*. Sparse signals are characterized by having only a few (but unknown) non-zero components. This leads to the question if signals can be sampled at a lower rate than predicted by the WKS theorem. Sampling at a lower rate would support to realize cheaper, smaller and less energy consuming analog-to-digital conversation.

This thesis compares and analyzes three different novel sub-Nyquist sampling systems: The *random demodulator* (RD), some methods using a framework termed *finite rate of innovation* (FRI) and the *modulated wideband converter* (MWC). In particular it is discussed, if these approaches outperform prevalent sampling strategies in practice.

Keywords: Whittaker-Kotelnikov-Shannon theorem, Analog-to-digital conversation, sampling, sub-Nyquist sampling, compressive sampling, spectral compressive sampling, finite rate of innovation, random demodulator, modulated wideband converter

Kurzfassung

Gewöhnliche Abtastsysteme basieren auf dem Whittaker-Kotelnikov-Shannon (WKS) Abtasttheorem, welches besagt, dass ein Signal mit mindestens der doppelten im Signal vorhandenen Frequenz abgetastet werden muss. Dies ist jedoch ein Grenzfall. Denn es gilt sowohl für (band begrenztes) weißes Rauschen als auch für Signale, die nur eine niedrige Informationsrate besitzen.

Viele Signale können mit beträchtlich weniger Parametern dargestellt werden als das Abtasttheorem vorhersagt. Dies ist durch Darstellung in einer optimalen Form (zum Beispiel durch eine Transformation wie der Fourier Transformation) möglich, in der das Signal spärlich besetzt (*sparse*) ist. Spärlich besetzte Signale bestehen lediglich aus wenigen (jedoch unbekannt) Koeffizienten. Dies führt zur Frage, ob solche Signale nicht mit einer geringeren Rate, als durch das Abtasttheorem vorhergesagt abgetastet werden können. Eine niedrigere Abtastrate würde die Realisierung billigerer, kleinerer und stromsparenderer Analog-zu-Digital Umsetzung ermöglichen.

Diese Diplomarbeit vergleicht drei neuartige Abtastsysteme: Den *random demodulator* (RD), einige Methoden, die auf dem *finite rate of innovation* (FRI) *framework* basieren, sowie den *modulated wideband converter* (MWC) und untersucht diese hinsichtlich ihrer Praxistauglichkeit.

Stichwörter: Whittaker-Kotelnikov-Shannon Theorem, Analog-zu-digital Umsetzung, Abtastung, sub-Nyquist Abtastung, compressive sampling, spectral compressive sampling, finite rate of innovation, random demodulator, modulated wideband converter

Acknowledgments

This thesis was written at the Signal Processing and Speech Communication Laboratory (SPSC) at Graz, University of Technology in 2010.

First I would like to express my gratitude to my advisor Christian Vogel for his advice and all the fruitful discussions. I really appreciate your great support and guidance during this thesis.

Furthermore I would like to thank the people of the laboratory, especially the master students in the DSP lab for their help and valuable comments.

Finally, I would like to thank my parents Alfred and Sophie and my girlfriend Manuela for all their support and understanding over the last years. Thank you for being here along the way during my studies.

Contents

1	Introduction	1
1.1	Motivation	3
1.2	Outline	7
2	Theoretical background	8
2.1	From Nyquist to sub-Nyquist	9
2.2	Compressive sampling	12
2.2.1	The restricted isometry property (RIP)	13
2.2.2	Random matrices	15
2.2.3	Sparsity and coherence	15
2.2.4	Recovery methods	17
2.2.5	Summary	18
2.3	The finite rate of innovation (FRI) framework	19
3	System model	21
4	The random demodulator (RD)	23
4.1	Working principle	23
4.1.1	The general model	23
4.1.2	Using an integrator as analog filter	24
4.1.3	Derivation of the minimum sampling rate	29
4.1.4	Generalization to arbitrary tone spacings	29
4.2	Implementation	30
4.2.1	Framework	30
4.2.2	Reconstruction	31
4.3	Evaluation of the random demodulator	33
4.3.1	Signal model	33
4.3.2	Sampling rate	35
4.3.3	Reconstruction performance	36
4.3.4	The integrator as anti-aliasing filter	38
4.3.5	Chipping sequence	39
4.3.6	Dependency on the signal model	42
4.3.7	The integrator	44

4.4	Spectral compressive sampling (SCS)	45
4.4.1	Recovery and sparse approximation algorithm for SCS	45
4.4.2	SCS via spectral estimation	46
4.4.3	Evaluation of SCS with the random demodulator	47
4.5	Conclusion	49
5	Sampling at the rate of innovation	50
5.1	Main idea and working principle	51
5.2	Why Diracs?	54
5.3	The basic case: Periodic stream of Diracs using the sinc kernel	55
5.3.1	Implementation using the annihilating filter	55
5.3.2	Stability of recovery algorithm	61
5.3.3	Error measures	61
5.4	Other recovery strategies	64
5.4.1	SVD approach	64
5.4.2	Subspace approach	66
5.4.3	Evaluation	66
5.5	Closely spaced Diracs	68
5.6	Butterworth filter	70
5.7	Gauss kernels	71
5.8	Spline kernels	72
5.8.1	Polynomial reproducing kernels	72
5.8.2	Exponential kernels	75
5.9	Conclusion	78
6	The modulated wideband converter (MWC)	79
6.1	Working principle	80
6.1.1	Compressive sampling for analog signals	80
6.1.2	The equivalent diagram of the MWC	83
6.1.3	Parameters of the sampling system	85
6.1.4	Reconstruction	87
6.1.5	Reduction of channels	88
6.2	Evaluation of the MWC	91
6.2.1	The MWC and Non-Blind sampling	92
6.2.2	Number of channels	92
6.2.3	Non-ideal low-pass filter	94
6.2.4	Non-ideal modulation sequences	95
6.2.5	Sub-Nyquist processing	96
6.2.6	Real-time processing	96
6.3	Conclusion	98

Contents

7	Comparison and summary	99
7.1	Properties	99
7.2	Summary	101
8	Conclusion	106
A	Derivation of the coefficients in the polynomial reproduction formula	108
B	Derivation of the coefficients in the exponential reproduction formula	112

List of Figures

1.1	Classical, ideal sampling setup	2
1.2	Historical trends of P	6
2.1	Example of a union of subspaces	10
2.2	Sampling of a K -sparse vector	13
3.1	Block diagram of a sub-Nyquist sampling device	21
4.1	Block diagram of the RD	23
4.2	Equivalent diagrams of the RD	25
4.3	Baseband of three different tones	26
4.4	Equivalence between the equivalent diagrams	27
4.5	Comparison of the sensing matrices	29
4.6	SIMULINK model of the RD	30
4.7	Reconstruction of non-multitone signals	34
4.8	Minimum sampling rate versus practical sampling rate	35
4.9	Running time of signal recovery	37
4.10	Input signal distorted by the integrator	38
4.11	Transfer function of the integrator	39
4.12	Ideal and low-rate interpolated chipping sequence	40
4.13	Reconstruction using a resampled low-rate signal as chipping sequence	40
4.14	Jitter applied to the chipping sequence	42
4.15	Slight drift from signal model	43
4.16	Effect of integration bias	44
4.17	Required sampling rate for SIHT	47
4.18	Computational load of spectral iterative hard thresholding (SIHT)	48
5.1	Block diagram of FRI	50
5.2	Basic principle of FRI reconstruction	54
5.3	Reconstruction performance when increasing B	57
5.4	Sampling of a signal consisting of two Diracs	58
5.5	Stability when number of Diracs K increases	60
5.6	Stability using annihilating filter	62
5.7	Number of failing amplitude recoveries	63

List of Figures

5.8	Estimation of number of Diracs K with SVD approach	65
5.9	Stability for all three recovery methods	67
5.10	Noise robustness	67
5.11	Required oversampling in the noisy case	68
5.12	Estimation of number of Diracs K with subspace approach	69
5.13	Closely spaced Diracs	69
5.14	Reconstruction performance with Butterworth filter	70
5.15	Stability with Gauss filter	72
5.16	Sampling Diracs using B-splines	74
5.17	Oversampling using B-splines	75
5.18	Stability using B-splines	76
5.19	Possible usage of E-splines	76
6.1	Block diagram of the MWC	79
6.2	Spectrum and vector $\mathbf{z}(f)$ of a multiband signal	81
6.3	Equivalent diagram of the MWC	83
6.4	Digital reconstruction of the MWC	88
6.5	The vector $\mathbf{z}(f)$ when $f_s = 4f_p$	89
6.6	Required number of channels	93
6.7	Effect of using non-ideal low-pass filters	95
6.8	Real-time capability of the MWC	97
A.1	Polynom reproduction using B-splines	111
B.1	Family of E-splines	113

List of Tables

4.1	Running times of reconstruction algorithms	32
7.1	Comparison of the sub-Nyquist sampling systems	100

Acronyms

ADC	analog-to-digital converter
CoSaMP	compressive sampling matching pursuit
CS	compressive sampling
CTF	continuous-to-finite converter
DAC	digital-to-analog converter
DFT	discrete Fourier transform
DSP	digital signal processor
DSSS	direct-sequence spread spectrum
DTMF	dual-tone multi-frequency signaling
ESPRIT	estimation of signal parameters via rotational invariant techniques
FFT	fast Fourier transform
FRI	finite rate of innovation
IC	integrated circuit
IDTFT	inverse discrete-time Fourier transform
IHT	iterative hard thresholding
IMV	infinite measurement vector
iRWLS	iteratively reweighted least squares
ISI	inter symbol interference
LFSR	linear feedback shift register
MMV	multiple measurement vector
MP	matching pursuit

MSE	mean squared error
MUSIC	multiple signal classification
MWC	modulated wideband converter
NMSE	normalized mean squared error
OFDM	orthogonal frequency-division multiplexing
OMP	orthogonal matching pursuit
RD	random demodulator
RIP	restricted isometry property
SCS	spectral compressive sampling
SeDuMi	self-dual-minimization
SIHT	spectral iterative hard thresholding
SMV	single measurement vector
SNDR	signal-to-noise-and-distortion ratio
SNR	signal-to-noise ratio
SVD	singular value decomposition
UWB	ultra-wide band
WKS	Whittaker-Kotelnikov-Shannon

List of Symbols

Nomenclature

\mathbf{a}	a vector \mathbf{a}
\mathbf{A}	a matrix \mathbf{A}
$x(t)$	an analog (time dependent) signal x
$x[n]$	a discrete signal x
$X[m]$	Fourier series of a signal x
$X(z)$	z -transform of a signal x
$(\cdot)^H$	hermitian of a matrix
$(\cdot)^\sharp$	Moore-Penrose pseudo inverse of a matrix: $\mathbf{A}^\sharp = (\mathbf{A}^H \mathbf{A})^{-1} \mathbf{A}^H$
*	convolution operator
\Re, \Im	real- and imaginary part
$(\cdot)^*$	complex conjugate
A'	variable used the same way as A but with different value
$\delta(t)$	Dirac-delta distribution ($\int_{-\infty}^{\infty} \delta(t) dt = 1$)
j	imaginary unit ($j = \sqrt{-1}$)
$\text{diag}(\cdot)$	diagonal matrix having the arguments in the main diagonale
$\langle x, y \rangle$	either L_2 ($\int_{-\infty}^{\infty} x^*(t)y(t) dt$) or ℓ_2 ($\sum_i x_i^* y_i$) inner product
$\ \mathbf{a}\ _0$	ℓ_0 pseudo norm (number of non-zero entries in \mathbf{a})
$\ \mathbf{a}\ _p$	ℓ_p norm of \mathbf{a} , i.e., $\ \mathbf{a}\ _p = (\sum_i x_i ^p)^{1/p}$
\mathcal{H}	Hilbert space

Common symbols

K	sparsity number of Diracs
L	upper bound on K , i.e., $K \leq L$
M	number of CS measurements number of branches in multichannel setup number of samples (may be decomposed as $M = 2M_0 + 1$)
N	size of full K -sparse CS vector (may be decomposed as $N = 2N_0 + 1$) number of samples order of B-spline/E-spline

Φ	CS measurement matrix, $M \times N$
Ψ	CS sparsity inducing matrix, $N \times N$
Θ	CS matrix ($\Theta = \Phi \cdot \Psi$), $M \times N$
f_N	Nyquist rate of input signal
T_N	inverse of Nyquist rate ($T_N = 1/f_N$)
f_s	low-rate sampling rate
T_s	low-rate sampling period ($T_s = 1/f_s$)
f_{tot}	total sampling rate (equal to f_s for RD and FRI)
T	period of a periodic signal
$x(t)$	input signal
$\hat{x}(t)$	reconstructed, analog signal
$y(t)$	signal prior to (low-rate) sampling stage (after preprocessing)
$\mathbf{y}(t)$	M sequences of $y(t)$ in a multichannel setup
$\mathbf{y}(f)$	$\mathbf{y}(t)$ in frequency domain
$y[n], y[m]$	samples of $y(t)$ (low-rate sample sequences)
$\mathbf{y}[n]$	M sequences of $y[n]$ in a multichannel setup
c_n, \mathbf{c}	amplitude(s) of sinusoid(s) weight(s) of Dirac(s)
$h(t)$	analog prefilter
$g(t)$	arbitrary pulse shape
$p_c(t), p_i(t)$	chipping/modulation sequences used in preprocessing stage
α	sign value of entries in the chipping sequences
\mathbf{A}	sign matrix (containing the values α)
\mathbf{F}	DFT matrix (where appropriate with shifted columns)
\mathbf{D}	diagonal matrix
\mathcal{V}	linear signal subspace
\mathcal{X}	non-linear signal subspace

Symbols used with the RD

Ω	set of frequencies contained in $x(t)$
\mathbf{H}	compression matrix
Δ	frequency spacing

Symbols used with FRI

t_k	time shifts of Diracs
ρ	rate of innovation (e.g., for a stream of Diracs $\rho = 2K/T$)
B	bandwidth of analog prefilter (related to ρ)

M_F	number of Fourier coefficients
$A(z)$	annihilating filter (in z -domain)
$a[m], \mathbf{a}$	coefficients of annihilating filter
z_k	roots of the annihilating filter (encodes t_k)
$\mathbf{W}_2, \mathbf{w}_1$	vector and matrix in a Toeplitz or Hankel system
$\mathbf{V}_2, \mathbf{v}_1$	vector and matrix in a Vandermonde system
$\beta(t)$	B-spline/E-spline kernel (analog prefilter)
$B(\omega)$	$\beta(t)$ in frequency domain

Symbols used with the MWC

$\mathbf{z}(f), \mathbf{z}[n]$	vector containing slices of the spectrum $X(f)$ of $x(t)$
B	maximum bandwidth of a band
f_i	carrier frequencies ($i = 1, \dots, K/2$)
T_p	period of $p_i(t)$
f_p	fundamental frequency of $p_i(t)$, $T_p = 1/f_p$
\mathcal{F}_s	frequency range of the low-rate ADCs
\mathcal{F}_p	frequency range of the modulation sequences
\mathcal{S}	indexes of non-zeros of a sparse vector (support, sparsity pattern)
$\Phi_{\mathcal{S}}$	matrix Φ having only columns contained in the set \mathcal{S}
$\mathbf{z}_{\mathcal{S}}$	vector \mathbf{z} having only entries contained in the set \mathcal{S}
\mathbf{U}	collection of sparse vectors (as columns) in an MMV system
\mathbf{V}	collection of measurement vectors (as columns) in an MMV system

Chapter 1

Introduction

Sampling is the process of converting analog signals to a sequence of digital numbers which can be processed by a computer. Devices which perform this conversation are called analog-to-digital converter (ADC). An ADC essentially takes values of a continuous signal (*samples*) which are discretized afterwards to finite precision numbers. There is hardly a modern electronic device without an integrated ADC. ADCs are required whenever real-world data must be transferred into a computer (for example when recording music, voice or movies, recording images or in reception of communication signals) and as such, they are a fundamental part of today's electronic devices.

The mathematical theory underlying the sampling process is strongly related to interpolation of functions. The earliest traces lead back to at least 1765 when Joseph-Louis Lagrange addressed the interpolation of periodic, bandlimited functions. In 1915, Edmund Taylor Whittaker published his work "On the functions which are represented by the expansions of the interpolation theory", which first addressed the interpolation of arbitrary, bandlimited functions [1]. From a practical point of view, the foundations date back to 1928 when Harry Nyquist explored how many pulses can be transmitted from a communication system with bandwidth B [2]. The term Nyquist rate (the minimum sampling rate) still reminds us of his pioneering work. A unified theorem, which describes how many samples are sufficient to describe an analog signal $x(t)$ uniquely, was first established by Wladimir Alexandrowitsch Kotelnikov in 1933 [3] and independently by Claude Elwood Shannon in 1949 [4]. The theorem, which is known today as the Whittaker-Kotelnikov-Shannon (WKS) sampling theorem, states the sampling rate must be at least twice as high as the highest frequency component in $x(t)$ [4]:

Theorem 1. *If a (finite-energy) function $x(t)$ contains no frequencies higher than W Hz, it is completely determined by giving its ordinates at a series of points spaced $\frac{1}{2W}$ seconds apart.*

Figure 1.1 shows the block diagram of a classical, ideal sampling setup. First, the analog signal $x(t)$ is filtered by an analog anti-aliasing filter to reject unwanted signal components. In practice, this is a low-pass filter. The next step is the actual sampling step, whose task is to convert the analog, continuous signal $y(t)$ to a sequence of numbers. In order to define the input signal $x(t)$ with the samples uniquely, the number of samples

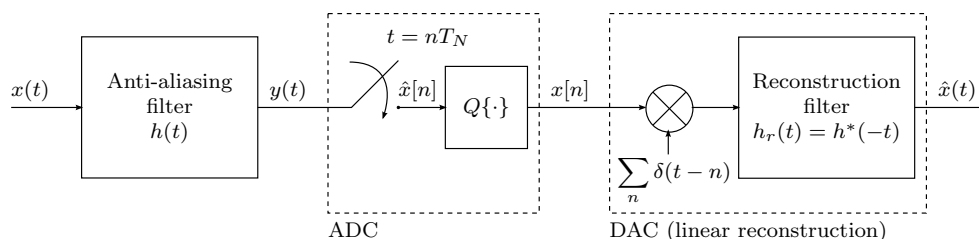


Figure 1.1: Classical, ideal sampling setup: The input signal is first filtered with an anti-aliasing filter $h(t)$ which is a low-pass filter, followed by uniform sampling which generates a sequence of numbers $\hat{x}[n]$. A quantizer Q converts the numbers to integer values. The reconstruction modulates the discrete sequence onto a Dirac stream which is finally low-pass filtered

per second must obey theorem 1. The number of samples an ADC can acquire per second is termed *sampling rate* and is the most important parameter of an ADC. The faster an ADC samples, the more expensive it is in terms of size, energy consumption and production costs. Current state-of-the-art ADCs can take samples up to 40 billion times per second (40 GSamples/s) [5]. Until now, the sequence consists of real numbers. In order to convert these to finite numbers computers can work with, a quantizer Q converts them to a sequence of numbers at finite precision. The quantizer is the only part in classical sampling, where information loss occurs because a real number is mapped to a finite number of bits per sample, for example 8 bit. The number of bits an ADC can convert is called *resolution* and is the second important parameter. Similar as for the sampling rate, properties such as size and energy consumption are related to the number of bits directly. In a common setup, the process is finished now and the numbers $x[n]$ are passed to a computer such as a digital signal processor (DSP) for further digital processing.

In case the original signal needs to be reconstructed, a digital-to-analog converter (DAC) is used, which generally is much easier to realize. First, the sequences are modulated by a stream of Dirac impulses in order to obtain an analog pulse stream. Finally, this signal is filtered with an analog reconstruction filter $h_r(t)$. Generally, the reconstruction filter $h_r(t)$ is the time-reversed, conjugate of the anti-aliasing filter $h(t)$. In case of bandlimited functions, both filters are ideal low-pass filters and therefore have a sinc function as an impulse response. Since the sinc function is real and symmetric, the reconstruction filter $h_r(t)$ is essentially the same as the anti-aliasing filter $h(t)$. It is important to note that the whole reconstruction process is *linear*: The multiplication with the Dirac comb is a linear operation, as well as the final filter operation. Assuming no quantization errors, ideal components and theorem 1, the reconstructed signal $\hat{x}(t)$ is equivalent to the original signal $x(t)$.

Usually there is desire for fast devices with a very high sampling rate, while simul-

taneously having a high resolution such as 16 bit or even 24 bit. The WKS sampling theorem often provides a *sufficient*, but by no means a *required* condition¹. It is easy to believe that a signal with a single sinusoid of 1 GHz does not require about two billions of samples per second to specify it uniquely since the signal can be described only by the three parameters amplitude, frequency and phase shift. Many scientists say that Shannon was a pessimist. Indeed, when the signal has a certain structure, the WKS theorem only provides an upper bound.

Many signals occurring in practice have a kind of structure which supports the signal to be described with considerably less parameters than with the WKS theorem. In general, signals can be described in some mathematical representation where the properties are expressed as coefficients. For example, bandlimited signals can be represented as a linear combination of trigonometric polynomials using the Fourier transform. An analog signal which has only a low number K of active coefficients is called *sparse analog signal*. For example, a dual-tone multi-frequency signaling (DTMF) signal is highly sparse in Fourier domain since it has only two active coefficients (frequencies). Other signal priors than sparsity, such as smoothness priors have been proposed [6].

The aim of this thesis is to analyze and compare three novel methods which enable sampling of sparse analog signals at a rate considerably lower than predicted by the WKS theorem. The practical realizability of such systems would enable cheaper and smaller sampling systems and would even allow to sample signals which can not be sampled nowadays.

1.1 Motivation

The WKS has been the heart of signal acquisition for over five decades. Until now, essentially all devices requiring signal acquisition obey this theorem. While there has been extensive research in the field of sampling over the last decades [7], the heart of any sampling device is still the WKS theorem. While the original sampling theorem was motivated from a rather practical point of view (sampling of communication signals with the use of readily available low-pass filters), following approaches offered a different point of view on sampling. These approaches for example describe the anti-aliasing filter as projection operator and introduce alternative basis functions such as splines [7]. Despite the advances in sampling theory, the requirement on the sampling rate still kept the same.

Recently, an alternative sampling theory termed compressive sampling (CS) or compressed sensing has been established, which has the potential to outperform the prevalent WKS theorem. It became widely known when Emmanuel Candès, David Donoho, Justin Romberg and Terence Tao published some pioneering papers [8, 9, 10]. Important spade-work was done in 2001 by Scott Chen, Michael Saunders and Donoho [11]. While this

¹This is not universally valid. For example, it does not apply to bandlimited white noise

novel theory was established only recently, the prehistory is much older. In the 1970ies, seismologists found out that they were able to construct images from reflective layers within the earth which did not obey the WKS theorem [12]. Moreover, the theory behind CS is far from new: It is, simply speaking, based on solving a system of underdetermined, linear equations.

Another related theory appeared even earlier than the celebrated CS around 2001 in the Ph.D. thesis of Pina Marzillano [13]. This theory termed finite rate of innovation (FRI) sampling is motivated by the observation that a stream of Diracs, although having theoretically infinite bandwidth, is characterized by a small set of parameters which can be estimated from a small set of Fourier coefficients.

The motivation for an alternative sampling system than the WKS theorem is manifold:

Sampling of non-bandlimited signals Using the WKS sampling theorem it is not possible to sample non-bandlimited signals (signals whose highest frequency component tends to infinity). However, many of those signals are not bandlimited but have a very simple structure, such as piecewise polynomials. Sampling such signals using the WKS theorem distorts the sampled signal.

Sparsity Many signals have a compact mathematical representation although they have a high bandwidth. Again this is true for piecewise polynomial signals, splines as well as for example a small set of frequencies distributed across a wide frequency range. Sampling of such signals using the WKS theorem is a waste of resources.

Avoid sample and discard Consider a digital camera acquiring images at high resolution of a few megapixels. Such a photograph requires 50 MB and more of memory. However, usually the image is saved as a JPEG image, resulting in a small sized file of a few hundred kB, without much information loss. The underlying compression method is based on representing the image in an appropriate basis, for example the discrete cosine transform or the wavelet transform, under which the image is sparse. In this sparse representation, coefficients having a value near zero are discarded. This observation is a key-concept in CS. Another example appears in the reception of communication signals: Due to analog modulation, an analog signal is sampled at a rate much higher than the information rate. Most parts of the sampled data are discarded after the acquisition process. It is a natural question whether it is really necessary to acquire this huge amount of data if it is discarded anyway or whether it is possible to acquire data directly in a compressed way.

The need for better ADCs Using novel sampling methods, it might be possible to sample signals with a certain structure but with a bandwidth exceeding todays capabilities, such as extremely high frequency signals. Furthermore, more and more signal processing

shifts to digital domain. Future trends like software defined radio or fourth generation mobile systems accelerate this process and need fast conversion as a gate to the analog world. The well known Moore's law predicts doubling of computational throughput about every two years whereas the performance of ADCs doubles only every 4.7 years [14]. Current ADC architectures are not able to catch up the requirements defined by modern applications.

Another technology which is in the need of novel sampling methods is ultra-wide band (UWB). UWB has been proposed as a technology for use in personal area networks and appeared even in the standard IEEE 802.15.3a. However, after several years of deadlock, the task group was dissolved and many vendors stopped research and production of UWB products. High implementation costs and performance problems are some of the reasons for limited success in consumer products [15].

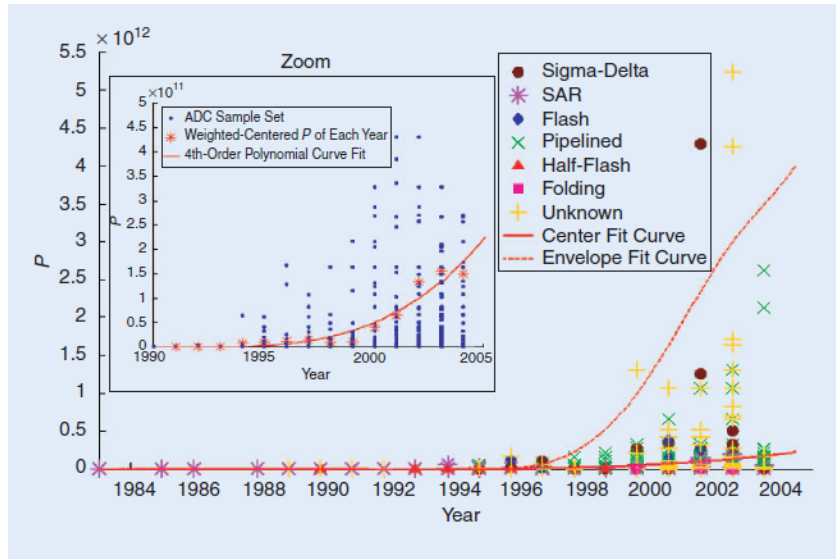
Sampling rate vs. resolution As discussed, the sampling rate and the resolution are the most important parameters of an ADC. Unfortunately, novel applications need fast conversion and good resolution at the same time. A widely used figure of merit is the product of these two numbers, defined as [16]:

$$P := 2^B \cdot f_s \tag{1.1}$$

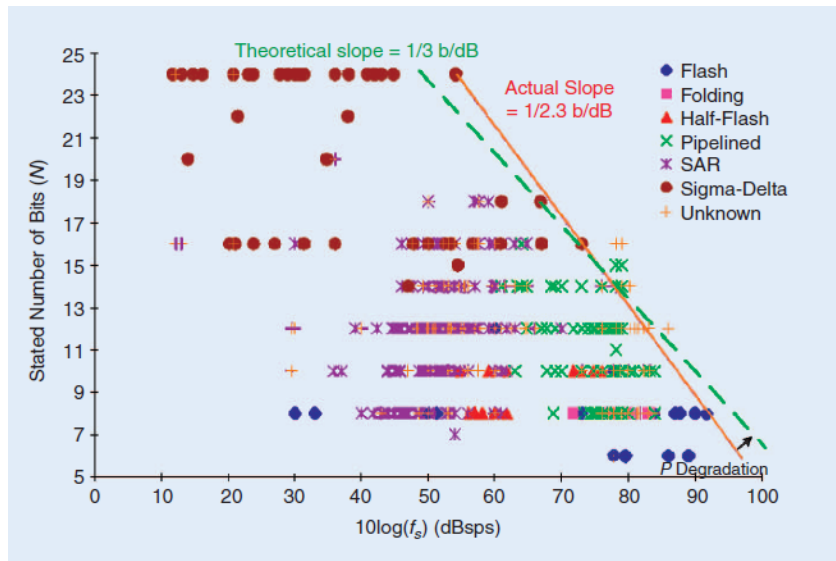
with B being the number of bits and f_s the sampling rate. Unfortunately, it is not possible to increase both parameters at the same time without limit (an ultimate limit was estimated using Heisenberg's uncertainty principle in [17]). In 1999, Walden predicted the very pessimistic trend that P stays nearly constant. Fortunately, this prediction turned out to be wrong as can be seen in figure 1.2(a), where P shows an exponential trend since 1994 (see the center fit curve in the zoomed version) [16].

Nevertheless, fast ADCs still have a low resolution and this trend is even continuing. Figure 1.2(b) shows the resolution versus the sampling rate for over 1000 ADCs over the past 20 years [16]. The green line shows the trend from Walden in 1999. The orange line shows the actual trend which indicates a degradation of P in the high rate regime. A possible solution to overcome the problem of limited resolution is the reduction of the sampling rate (by assuming a sparse analog input signal) which provides higher resolution, especially in high-rate ADCs. Further digital reconstruction recovers the original signal.

Trading analog vs. digital Since Moore's law is faster than ADC improvement, one option to overcome the ADC bottleneck is to trade analog processing with digital processing. Such examples are ADCs with digital assisted circuits [18]. Sparse signals can help in this process because they support to incorporate prior knowledge: An ADC could sample at a low rate but high resolution and use computational power in digital domain to reconstruct the signal.



(a)



(b)

Figure 1.2: Sampling rate vs. resolution (a) historical trend of P . Although predicted to be constant by Walden, it exhibits an exponential behavior [16] (b) Although P exhibits an exponential decay, there is an increase of resolution in the low-rate regime and a decrease in the high-rate regime [16]

Another important property is power dissipation. This property is especially important for mobile applications which are the leading devices requiring fast ADCs. In the last years, the majority of the designs focused on an optimization of power dissipation, rather than maximizing the bandwidth [18].

While the focus on power efficiency has led to an average power reduction in ADCs every two years, this reduction is still small compared to the reduction in digital logic: While the power dissipation in logic circuits improves 300 times in ten years, it improves in ADCs only 32 times within the same period. For example, one conversation with an signal-to-noise-and-distortion ratio (SNDR) of 90 dB can toggle over two million logic gates [18]. An interesting question is if current state-of-the-art sampling algorithms for sparse analog signals can profit from this gap in digital computation power.

1.2 Outline

This thesis is organized as follows: The next chapter provides a small introduction about the theoretical foundations. In particular, the two most important frameworks used in this thesis, the CS and FRI frameworks are introduced.

Chapter 3 describes the sampling setup of the sub-Nyquist sampling systems. The possible input signals and the three basic building blocks (the preprocessing, low-rate sampling and non-linear reconstruction) are introduced. Furthermore, the nomenclature is established.

Chapter 4 covers the first sub-Nyquist sampling system, the random demodulator (RD), which directly applies principles from CS field. The device samples signals being sparse in frequency domain, that is, signals consisting only of a few frequency components.

Chapter 5 addresses a series of concepts which make use of the FRI framework. These devices sample streams of Diracs. In the basic case, these signals are periodic, but with advanced sampling methods involving compactly supported kernels, it is possible to sample aperiodic streams of Diracs.

Chapter 6 covers the modulated wideband converter (MWC). The MWC is a sub-Nyquist sampling system which is much based on the classical sampling theory while using key-ideas from the CS framework. The device samples multiband signals, that is, signals consisting a series of concurrently narrow-band transmissions.

Chapter 7 provides an overview on the main aspects of all discussed systems and discusses the advantages and disadvantages.

Finally, conclusions are given in chapter 8.

Chapter 2

Theoretical background

The reconstruction formula in the WKS sampling theorem is given by [4]:

$$x(t) = \sum_{n=-\infty}^{\infty} c_n \operatorname{sinc}(t - nT_N) \quad (2.1)$$

where the coefficients c_n correspond to the samples $x[n]$ and T_N is the sampling period (the inverse of the sampling rate f_s). While (2.1) is widely interpreted as interpolation formula [19], its meaning is wide-ranging: It can be interpreted as a subspace where all bandlimited functions live in and which can be uniquely described by the discrete sequence c_n . In the concrete case, the subspace is spanned by shifted versions of a single generator, the sinc function. Such a subspace is called a linear, shift invariant subspace.

This concept can be generalized to arbitrary generators $\psi_n(t)$. A vector space called \mathcal{V} , which is spanned by the generators $\psi_n(t)$ can be defined as:

$$\mathcal{V} := \left\{ x(t) = \sum_n c_n \psi_n(t) : c_n \in \ell_2 \right\} \quad (2.2)$$

In practice, $x(t)$ is assumed to have finite energy, that is, it is part of a Hilbert space $x(t) \in L_2$. Furthermore, in order for the process to be stable, the basis functions ψ_n are chosen to form a Riesz basis [7]. In this case, the coefficient sequence has finite energy as well, that is, $c_n \in \ell_2$. With $\psi_n = \operatorname{sinc}(t - nT_N)$ and $n \in \mathbb{Z}$, the vector space \mathcal{V} describes the class of bandlimited functions and we are back to (2.1). With (2.2) in mind, the components of the sampling setup described in figure 1.1 can be interpreted as follows:

The anti-aliasing filter The goal of the anti-aliasing filter is to produce a bandlimited function, that is, a function which can be described by the vector space spanned by (2.2). Suppose $x(t) \in L_2$ is not bandlimited (or the highest frequency component exceeds $f_N/2$). Then the anti-aliasing filter projects $x(t)$ onto the subspace spanned by (2.2) such that the projection yields a minimum-error approximation. The projection $P_{\mathcal{V}} : L_2 \rightarrow \mathcal{V}$ can be written as [7]:

$$P_{\mathcal{V}}x(t) = \sum_n \langle x(t), \psi_n(t) \rangle \cdot \psi_n(t) \quad (2.3)$$

In this sense, the anti-aliasing filter acts as a projection operator. When the input signal $x(t)$ is already in the subspace \mathcal{V} , the operator $P_{\mathcal{V}}$ is without effect.

The ADC After the anti-aliasing filter, $x(t) \in \mathcal{V}$ and therefore the coefficients c_n uniquely describe $x(t)$. When ignoring all non-ideal effects (such as the quantizer Q), the sample sequences $x[n]$ now uniquely describe the input signal $x(t)$.

The DAC The act of the DAC can be interpreted as a reconstruction operator R which converts the coefficients $c_n \in \ell_2$ back to L_2 : $R : \ell_2 \rightarrow L_2$.

It is important to note that the whole sampling setup described so far is linear and the symbol \mathcal{V} denotes a linear signal subspace. Under certain practical assumptions, it is possible to realize a sampling setup, as long as it keeps linear. An example for a sampling setup consisting of multiple, shift invariant generators based on a multichannel setup is given in [20].

2.1 From Nyquist to sub-Nyquist

Suppose the signal $x(t)$ can be described in a finite dictionary consisting of N elements:

$$x(t) = \sum_{n=1}^N c_n \psi_n(t) \quad (2.4)$$

In general, this corresponds to the setup described above which is linear. With N elements, this is an N -dimensional subspace. Now assume that only $K \ll N$ generators ψ_n (or entries in the sequence c_n) are active. To emphasize this setup, rewrite (2.4) as

$$x(t) = \sum_{n \in \Omega} c_n \psi_n(t), \quad \Omega \subset \{1, \dots, N\}, \quad |\Omega| = K \quad (2.5)$$

where the set Ω contains the indexes of the K active coefficients. When the set Ω is known, (2.5) consists of only K elements and therefore the subspace is only K -dimensional. When it is only known that there are K active generators but the set Ω itself is unknown, one option is still to describe the signal using N generators with (2.4). However, since $K \ll N$, there is hope to sample such a signal with fewer requirements than predicted

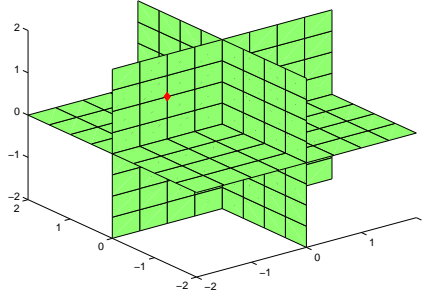


Figure 2.1: Example of a union of subspaces. In this example $N = 3$ and $K = 2$. The two-dimensional planes form three possible (linear) subspaces in which the signals may reside. For example, the red dot is the 2-sparse signal $(-1, 0, 1)$. The union over these subspaces forms a non-linear set in a three-dimensional space

by the WKS theorem. To describe this setup mathematically, denote one possible set Ω with K entries as Ω_k . The class of signals obeying the model $K < N$ can be described in a signal model \mathcal{X}_1 as follows:

$$\mathcal{X}_1 := \left\{ x(t) = \bigcup_{k=1}^{\binom{N}{K}} \sum_{i \in \Omega_k} c_i \psi_i(t) : c_i \in \ell_2 \right\} \quad (2.6)$$

where k iterates over all possible permutations of K active entries in $x(t)$. The signal space defined by \mathcal{X}_1 perfectly describes all signals of the form (2.4) with only K non-zero coefficients. The signals described by (2.6) lie in a K -dimensional, linear subspace. It is, however, not known in advance, in which of the $\binom{N}{K}$ subspaces it lies. The signal model is thus called a *union of subspaces*. However, it is not linear any more: The definition of \mathcal{X}_1 includes only signals with a maximum of K active coefficients. The summation of two signals from \mathcal{X}_1 by contrast, may have $2K$ active coefficients and is thus not included in \mathcal{X}_1 . The signal model is visualized in figure 2.1. In the example, $N = 3$ and $K = 2$ so that there are three possible linear, two-dimensional subspaces. The union over these subspaces form a non-linear set in the three-dimensional space. The red dot corresponds to a possible signal described by $c_0 = -1$, $c_1 = 0$ and $c_2 = 1$.

Another example of a signal model consisting of a union of linear subspaces appears when considering a signal $x(t)$ consisting of K pulses with pulse shape $h(t)$, weights c_k and time shifts t_k :

$$x(t) = \sum_{k=1}^K c_k h(t - t_k) \quad (2.7)$$

When the time shifts t_k are left constant, the signal model (2.7) defines a linear, K -dimensional subspace with the coefficients c_k . With arbitrary time shifts t_k , a union of all possible values of t_k forms a non-linear signal model \mathcal{X}_2 (a similar model is true when the signal $x(t)$ is periodic with period T). This time, define with $\Omega_i \subset \mathbb{R}$, $|\Omega_i| = K$ one particular set of possible time delays t_k in order to form the signal space:

$$\mathcal{X}_2 := \left\{ x(t) = \bigcup_{i=1}^{\infty} \sum_{k \in \Omega_i} c_k h(t - k) : c_k \in \ell_2 \right\} \quad (2.8)$$

The last important example covers multiband signals which consist of a maximum of K bands. Suppose that the spectrum is divided into N equally spaced spectrum cells and that the bands are positioned such that only K out of N cells contain signal energy. One way is to sample in a linear subspace such as (2.1) at the Nyquist rate again. However, to make use of the sparsity knowledge, the signals again define a union of linear subspaces model which can be described as [20]:

$$\mathcal{X}_3 := \left\{ x(t) = \bigcup_{k=1}^{\binom{N}{K}} \sum_{n \in \mathbb{Z}} c_{n,k} \psi_k(t - nT_N) : c_{n,k} \in \ell_2 \right\} \quad (2.9)$$

In this configuration, the generators $\psi_k(t)$ are weighted and modulated sinc functions.

To summarize, all the signal examples above are defined within a union of linear subspaces [21]:

$$\mathcal{X} := \bigcup_{k \in \Gamma} \mathcal{V}_k, \quad \mathcal{V}_k \in \mathcal{H} \quad (2.10)$$

A signal model described by a structure like (2.10) is non-linear. It can be regarded as a generalization of (2.2). A novel theory which covers sampling systems in a union of subspaces has been established [20, 21]. Using this theory, it has been shown, that the sampling requirements in a union of subspaces indeed are lower in many cases than predicted by the WKS theorem. It even makes it possible to determine stability bounds or the existence of an invertible sampling operator [21]. However, a comprehensive theory is still missing.

In this thesis, a sampling system which supports to sample a signal below the Nyquist rate (i.e., $f_s < f_N$) is called *sub-Nyquist* sampling system. All sampling systems covered by this thesis obey the union of subspaces model: The RD (chapter 4) model \mathcal{X}_1 , the FRI framework (chapter 5) model \mathcal{X}_2 , and the MWC (chapter 6) model \mathcal{X}_3 . The non-linearity of those models is an inherent problem which requires, in contrast to the WKS theorem, non-linear reconstruction methods and special preprocessing. In general, the methods are complex and very specialized on a specific signal model.

For all methods covered by this thesis, the preprocessing is achieved by using additional filters, chipping sequences, modulators and signal splitters and the reconstruction by a non-linear algorithm which highly depends on the underlying signal model.

The rest of this chapter provides a short introduction on two important frameworks which can be used to solve a sampling problem with signals lying in a union of subspaces, in particular \mathcal{X}_1 to \mathcal{X}_3 .

2.2 Compressive sampling

Suppose a discrete signal $x[n]$ of length N . A typical strategy to sample such a signal is to take pointwise samples at the Nyquist rate which corresponds to pick out each element in $x[n]$, resulting in the length- N sample sequence $y[n] = x[n]$. The signal $x[n]$ as well as the samples $y[n]$ can be regarded as length- N vectors \mathbf{x} and \mathbf{y} . The sampling process can be represented using a sampling matrix Φ :

$$\mathbf{y} = \Phi \cdot \mathbf{x} \tag{2.11}$$

In the case described before, the matrix Φ corresponds to the identity matrix \mathbf{I} . Now suppose the vector \mathbf{x} has only $K \ll N$ active components. In this case, \mathbf{x} is called a K -sparse vector. Sampling this vector with the operator $\Phi = \mathbf{I}$ results in a waste of resources since most of the time, only zero values are being sampled. A generalization of this approach is not just to pick single samples but rather linear combinations of the input signal. In this case, the matrix Φ is not the identity matrix any more and it is called the *sensing matrix*.

Since \mathbf{x} contains only K non-zero values, one might ask if it is possible to use fewer measurements than N , i.e., $M < N$ linear measurements on \mathbf{x} . In this case, the measurements are termed *compressed measurements*. Indeed, removing the zero elements from \mathbf{x} , the sampling operator can be reduced to a $M \times K$ matrix. If this matrix has full rank (where $M \geq K$ is a necessary condition), it is invertible. This is explained in figure 2.2. If it is known that \mathbf{x} contains only K non-zero values, the sensing matrix Φ can be designed so that each possible $M \times K$ matrix is left invertible. However, this strategy does not make sure that the measurements \mathbf{y} are unique for one (and only one) K -sparse vector \mathbf{x} . Suppose a second K -sparse vector \mathbf{x}' . The difference $\mathbf{x} - \mathbf{x}'$ is

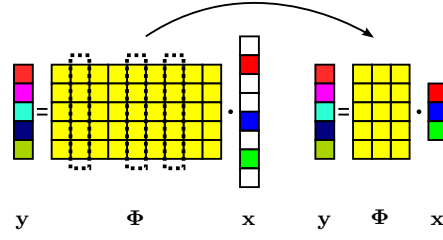


Figure 2.2: Sampling of a K -sparse vector \mathbf{x} : The K -sparse vector can be collapsed to a linear $M \times K$ system which is invertible as long as $M \geq K$. Additionally, when all $M \times 2K$ submatrices of Φ are invertible, each measurement \mathbf{y} is unique for any K -sparse vector \mathbf{x}

generally $2K$ -sparse, so to make sure that \mathbf{y} is unique for \mathbf{x} or \mathbf{x}' , the matrix Φ intuitively needs to be designed such that each $M \times 2K$ submatrix has full rank. Since under this assumption each measurement \mathbf{y} corresponds to a unique K -sparse \mathbf{x} , this vector can be found by iterating over all possible K -sparse vectors and seek for the one which explains the measurements \mathbf{y} . The ℓ_0 pseudo norm counts the number of non-zero entries in a vector so that this problem formally may be written as:

$$\min_{\mathbf{x}} \|\mathbf{x}\|_0 \quad \text{s.t.} \quad \mathbf{y} = \Phi \cdot \mathbf{x} \quad (2.12)$$

The approach until now has two severe shortcomings: First, how to find a matrix Φ (with $M \ll N$) where all possible $\binom{N}{2K}$ submatrices have full rank? Second, solving (2.12) requires iterating over all possible K -sparse vectors \mathbf{x} which is known to be NP-hard. Therefore it is not applicable to real-world applications.

2.2.1 The restricted isometry property (RIP)

To quantify the properties a matrix Φ needs to satisfy for stable and unique reconstruction of a K -sparse vector, the RIP is introduced. The RIP is also known as the uniform uncertainty principle (UUP).

Definition 1. For each integer $K = 1, 2, \dots$ define the isometry constant δ_K of the matrix Φ as the smallest number such that

$$(1 - \delta_K) \|\mathbf{x}\|_2^2 \leq \|\Phi \cdot \mathbf{x}\|_2^2 \leq (1 + \delta_K) \|\mathbf{x}\|_2^2 \quad (2.13)$$

holds for all K -sparse vectors \mathbf{x} [22].

Using definition 1, an important property of a matrix Φ , the RIP, can be defined as follows:

Definition 2. A matrix Φ is said to obey the RIP of order K when $\delta_K < 1$, in particular, when δ_K is not too close to 1 [23].

The interpretation of definition 1 and 2 is that a matrix Φ obeying the RIP of order K preserves the Euclidean norm of any K -sparse vector. Alternatively: All subsets of K columns taken from the matrix Φ are nearly orthogonal (they can not be exactly orthogonal since the matrix has more columns than rows) [23].

Theorem 2. The measurements \mathbf{y} are unique for a particular K -sparse vector \mathbf{x} using the measurements $\mathbf{y} = \Phi \cdot \mathbf{x}$ if Φ obeys the RIP of order $2K$. The solution may be found via (2.12).

Proof. Suppose there would exist a different K -sparse vector \mathbf{x}' which additionally explains the measurements \mathbf{y} . The difference $\mathbf{x} - \mathbf{x}'$ is generally $2K$ -sparse and since they correspond to the same measurements, they are in the null-space of Φ , that is

$$\Phi \cdot (\mathbf{x} - \mathbf{x}') = 0$$

However, this contradicts the assumption that Φ obeys the RIP of order $2K$ because if there exists a $\delta_{2K} < 1$, this can not be zero:

$$(1 - \delta_{2K})\|\mathbf{x} - \mathbf{x}'\|_2^2 \leq \|\Phi \cdot (\mathbf{x} - \mathbf{x}')\|_2^2 \leq (1 + \delta_{2K})\|\mathbf{x} - \mathbf{x}'\|_2^2$$

Therefore \mathbf{x} is unique for the measurements \mathbf{y} and may be found by exhaustive search through (2.12). \square

However, a more important result of CS is that \mathbf{x} can be found from \mathbf{y} in polynomial time if the following requirement is met:

Theorem 3. Assume that \mathbf{x} is an arbitrary vector and \mathbf{x}_K the best K -sparse approximation to \mathbf{x} . Assume that $\delta_{2K} < \sqrt{2} - 1$ and that \mathbf{y} are the measurements for the signal \mathbf{x} . Then the solution \mathbf{x}' to the optimization problem

$$\min_{\mathbf{x}'} \|\mathbf{x}'\|_1 \quad \text{s.t.} \quad \mathbf{y} = \Phi \cdot \mathbf{x}' \tag{2.14}$$

obeys

$$\begin{aligned} \|\mathbf{x}' - \mathbf{x}\|_1 &\leq c\|\mathbf{x} - \mathbf{x}_K\|_1 \\ &\text{and} \\ \|\mathbf{x}' - \mathbf{x}\|_2 &\leq cK^{-1/2}\|\mathbf{x} - \mathbf{x}_K\|_2 \end{aligned}$$

for some constant c explicitly given in [22]. In particular, if \mathbf{x} is K -sparse, the recovery is exact.

Proof. See the proof given in [22] □

The implications of theorem 3 are two-fold. First, the most important one, if Φ obeys the RIP with $\delta_{2K} < \sqrt{2} - 1$, then (2.12) is equivalent to (2.14). The latter can be solved using polynomial time recovery algorithms. Second, theorem 3 guarantees that recovery is exact for K -sparse signals. However, it does not apply only to K -sparse signals but to any signal \mathbf{x} . In this case, the K largest entries of \mathbf{x} are recovered.

2.2.2 Random matrices

Until now, no concrete details on the structure of Φ have been given. In particular, it is important how many rows M of Φ are enough for theorem 3, since this number corresponds to the sampling rate. In practice, verification if a particular matrix Φ obeys the RIP is infeasible since it requires to test (2.13) for all possible K -sparse vectors \mathbf{x} .

The second important contribution of CS is that a wide set of random matrices have been shown to obey the RIP (with $\delta_{2K} < \sqrt{2} - 1$). In particular, the following random matrices obey the RIP with “overwhelming” probability provided that the number of rows M follows

$$M \geq cK \log(N/K) \ll N \tag{2.15}$$

with c being a small constant, depending on the specific setup [23]:

- the entries in Φ are drawn uniformly at random.
- the entries in Φ are drawn from a normal distribution.
- the entries in Φ are drawn from a Rademacher or Bernoulli distribution.

The sampling methods in this thesis especially make use of the latter one since these matrices consist only of sign patterns ± 1 . These sign patterns easily can be generated in hardware using linear feedback shift registers (LFSRs).

2.2.3 Sparsity and coherence

In general, the theory underlying CS is based on two main principles:

1. sparsity
2. coherence

The discussion so far covered the former point: reconstruction of a K -sparse signal \mathbf{x} . However, what is the link to sub-Nyquist sampling? Consider again the analog signal $x(t)$ in (2.4). Suppose that this can be written in a discrete representation as follows:

$$x[n] = \sum_{i=1}^N c_i \psi_i[n] \quad (2.16)$$

or using matrix-/vector notation:

$$\underbrace{\begin{bmatrix} x[1] \\ x[2] \\ \vdots \\ x[N] \end{bmatrix}}_{\mathbf{x}} = \underbrace{\begin{bmatrix} \psi_1[1] & \psi_2[1] & \cdots & \psi_N[1] \\ \psi_1[2] & \psi_2[2] & \cdots & \psi_N[2] \\ \vdots & \ddots & \ddots & \vdots \\ \psi_1[N] & \psi_2[N] & \cdots & \psi_N[N] \end{bmatrix}}_{\mathbf{\Psi}} \cdot \underbrace{\begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_N \end{bmatrix}}_{\mathbf{s}} \quad (2.17)$$

When there are only $K \ll N$ active coefficients in (2.17), the coefficient vector \mathbf{s} is K -sparse. While \mathbf{x} is not sparse, it is said to be K -sparse with respect to the basis $\mathbf{\Psi}$. The matrix $\mathbf{\Psi}$ has the dimensions $N \times N$ and acts as the sparsity-inducing matrix. It is called *representation matrix*. With use of (2.17), (2.11) can be written as:

$$\mathbf{y} = \mathbf{\Phi} \cdot \mathbf{x} = \mathbf{\Phi} \cdot \mathbf{\Psi} \cdot \mathbf{s} \quad (2.18)$$

This is where *coherence* enters the play. The coherence μ between two bases $\mathbf{\Phi}$ and $\mathbf{\Psi}$ is defined as [23]:

$$\mu(\mathbf{\Phi}, \mathbf{\Psi}) := \sqrt{N} \cdot \max_{1 \leq i, j \leq N} |\langle \phi_i, \psi_j \rangle|, \quad \mu \in [1, \sqrt{N}] \quad (2.19)$$

Another prominent result of CS is that the sparse vector \mathbf{s} in (2.18) can be recovered as well using (2.14) [24]:

Theorem 4. *Suppose \mathbf{x} is K -sparse with respect to the basis $\mathbf{\Psi}$ and the measurements \mathbf{y} have been taken using $\mathbf{y} = \mathbf{\Phi} \cdot \mathbf{x}$. The sparse coefficients \mathbf{s} can be recovered using (2.14) with probability greater than $1 - \delta$ if the number of measurements M obeys*

$$M \geq c \cdot \mu(\mathbf{\Phi}, \mathbf{\Psi})^2 \cdot K \cdot \log(N/\delta) \quad (2.20)$$

with $c > 0$ being a specific constant.

Proof. See the original paper [24] □

Coherence measures the similarity or the correlation between the representation basis Ψ and the sensing matrix Φ . Equation (2.20) shows that recovery performs better, as the coherence decreases. The intuitive meaning behind the coherence is that the sensing waveforms should be uncorrelated largely with the waveforms being sampled. While the waveforms to be sampled are concentrated locally, the sensing waveforms should be spread out globally (and the other way round). For example, suppose sampling a sparse vector \mathbf{x} . This can be regarded as a stream of Diracs. The usage of sensing waveforms consisting of Diracs itself (such as in traditional, pointwise sampling) would require a large amount of samples because the samples contain only information when a sampling spike hits a signal spike. However, most of the samples will be zero. This conforms with (2.19) which predicts the maximum coherence when the two bases are the same. On the other hand, random waveforms have a small coherence with the Diracs. Since these waveforms are global, each sample will pick up a little information about each Dirac (encoded in its linear combination).

In general, it is possible to use theorem 4 directly. It has, however, some shortcomings compared to theorem 3. First, it includes the probability of failure. Despite δ being small, the number of samples in (2.20) directly depends on it. Theorem 3 by contrast always works, once a matrix Φ satisfying the RIP has been set up. Second, theorem 3 is able to deal with *all* signals \mathbf{x} while theorem 4 only deals with K -sparse signals. And finally, theorem 3 enables robust recovery from noisy measurements (see [23]).

With the $M \times N$ matrix $\Theta = \Phi \cdot \Psi$, (2.18) can be written as

$$\mathbf{y} = \Theta \cdot \mathbf{s} \tag{2.21}$$

Fortunately it has been shown that the multiplication of any fixed basis Ψ with a random matrix Φ obeys the RIP as well. The random measurements are *universal* in the sense that the sparsity inducing matrix Ψ is only needed in the recovery process [25]. For this reason, (2.21) can be solved directly using theorem 3.

2.2.4 Recovery methods

Nowadays, there exists a rich selection of solving methods for CS problems. Generally, the solving methods can be split into two categories:

1. Methods based on optimization.
2. Methods based on greedy algorithms.

Usually the former class works better for arbitrary signals but the computational load is higher. The latter class usually is faster and works better for ultra-sparse signals.

Solvers based on optimization

The original work in CS covers reconstruction using the ℓ_1 -norm via (2.14). This reconstruction method is called basis pursuit (BP) and is of utmost importance because on the one hand, it was a vital part of CS development to show the equivalence to the ℓ_0 -norm. On the other hand, ℓ_1 minimization can be recast as a linear program (LP) which can be solved in polynomial time [11]:

$$\min_{\mathbf{x}'} \mathbf{c}^T \mathbf{x}' \quad \text{s.t.} \quad \mathbf{A} \mathbf{x}' = \mathbf{b}, \quad \mathbf{x}' \geq 0 \quad (2.22)$$

$$\mathbf{A} = (\Phi, -\Phi), \quad \mathbf{b} = \mathbf{y}, \quad \mathbf{c} = (1; 1), \quad \mathbf{x}' = (\mathbf{u}; \mathbf{v}), \quad \mathbf{x} = \mathbf{u} - \mathbf{v} \quad (2.23)$$

Another option can deal with noise and can be recast as a second order cone program (SOCP) [26]:

$$\min_{\mathbf{x}} \|\mathbf{x}\|_1 \quad \text{s.t.} \quad \|\mathbf{y} - \Phi \cdot \mathbf{x}\|_2 \leq \epsilon \quad (2.24)$$

Other examples include basis pursuit with denoising (BPDN), the Dantzig selector, total variation methods [26] or iteratively reweighted least squares (iRWLS) [27]. The CVX package [28] offers a convenient interface for those optimization problems.

Solvers based on greedy algorithms

The second class of solvers works in a greedy fashion. In order to understand the principle behind those algorithms, take a look at figure 2.2 again. For each measurement \mathbf{y} and the corresponding K -sparse \mathbf{x} , there are only K columns of Φ involved. Greedy algorithms iterate over all columns of Φ and multiply each column by the measurement vector \mathbf{y} . The column with the largest result contributes most to \mathbf{y} and the corresponding entry in \mathbf{x} is added to the estimation. The whole procedure iterates until convergence. Popular variations are the matching pursuit (MP), orthogonal matching pursuit (OMP), stagewise orthogonal matching pursuit (StOMP), compressive sampling matching pursuit (CoSaMP) and thresholding methods.

2.2.5 Summary

To summarize, CS is a powerful framework for sub-Nyquist sampling. The underlying theory is based on sparsity (it is possible to represent data in an appropriate basis using only K out of N coefficients) and coherence (the representation and the sensing bases should be as less similar as possible). Randomness plays an important role because it is difficult to prove the RIP for a given sensing matrix.

Solving CS problems requires the sparse vector \mathbf{x} to be restored fully. This might be a problem if N is very large. It is mainly concerned with discrete signals where it proved as a good tool for image acquisition. For example, a digital camera consisting of a single pixel and making use of CS principles has been published [29]. An analog signal $x(t)$ in turn needs a discrete representation such as in (2.17) since an analog signal has infinite cardinality.

2.3 The finite rate of innovation (FRI) framework

FRI is a framework which can be used to solve signals which live in a union of subspaces as described by (2.8).

The name results from a key concept underlying the framework: It deals with signals having a finite rate of information, either in a local or a global sense. For example, take a look at (2.7). The signal $x(t)$ consists of a finite sum. Assuming the pulse shape $h(t)$ is known, it is completely characterized by a set of $2K$ parameters: K weights c_k and K time shifts t_k . Define signals with finite rate of innovation as follows [30]:

Definition 3. *A signal with a finite rate of innovation is a signal which can be represented in the parametric representation*

$$x(t) = \sum_{k \in \mathbb{Z}} \sum_{r=1}^R c_{k,r} g_r(t - t_k) \quad (2.25)$$

and has a finite value of ρ :

$$\rho = \lim_{\tau \rightarrow \infty} \frac{1}{\tau} C_x \left(-\frac{\tau}{2}, \frac{\tau}{2} \right) \quad (2.26)$$

The functions $g_r(t)$ are known pulse shapes and the function $C_x(a, b)$ counts the degrees of freedom in the interval $[a, b]$.

In the example (2.7), the rate of innovation is finite (since there are only K pulses) and given by $\rho = 2K/\tau$ with τ such that it includes all time shifts t_k .

Another interesting example appears when looking at the interpolation formula of the WKS theorem in (2.1). The only degree of freedom are the coefficients c_k . Since there is only one coefficient in each sampling period T_N , the rate of innovation is given by $\rho = 1/T_N = f_N$ and equal to the Nyquist rate. A possible interpretation is that the WKS theorem works because the signal has a finite rate of innovation rather than it is bandlimited [30].

The reason for introducing the definition of the innovation rate ρ is the hope being able to sample signals, which have a finite rate of innovation, directly at their rate of innovation. While this is possible for signals in a shift invariant structure such as (2.1), there is no known way to reconstruct signals from a union of subspaces in general.

However, Vetterli et al. showed that a limited selection of such functions can be sampled and reconstructed perfectly at their rate of innovation [31, 30]. This functions include periodic and aperiodic streams of Diracs and signals which can be reduced to a stream of Diracs such as pulses, nonuniform splines or piecewise polynomials.

Since the sampling and reconstruction process of FRI signals highly depends on their structure, the introduction to the working principle is given in chapter 5.

Chapter 3

System model

All three sampling methods covered by this thesis consist of three basic blocks: The preprocessing block, the sampling block and the reconstruction block. The block diagram is shown in figure 3.1. The sampling setup of the Whittaker-Kotelnikov-Shannon (WKS) theorem is shown in figure 1.1 and is taken as a reference. The most important difference is that the sampling block in figure 1.1 samples at the Nyquist rate f_N whereas the sampling block in figure 3.1 samples with a total sampling rate $f_{\text{tot}} < f_N$.

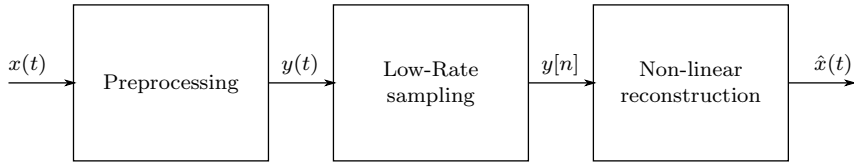


Figure 3.1: Block diagram of a sub-Nyquist sampling device

Input signal While the input signal $x(t)$ in the WKS case belongs to a linear signal subspace (2.1), the input signals for sub-Nyquist sampling systems belong to the non-linear signal model (2.10). In particular, $x(t)$ is one of the following three input signals:

1. A signal consisting of N total frequencies where only $K \ll N$ are active. Usually, $N = f_N$. The signal can be regarded as frequency sparse.
2. A signal consisting of K Diracs, either finite-length or periodic with T . The signal can be regarded as a signal which is sparse in time domain.
3. A multiband signal with $K/2$ transmissions of bandwidth B where each band is simulated by modulating a sinc. All bands are within $[-f_N/2, f_N/2]$. This signal can be regarded as a block-sparse signal in the frequency domain.

Preprocessing In the WKS case, the preprocessing consists of a simple low-pass filter which forces the signal to belong to the signal model (2.1). This is similar for the finite rate of innovation (FRI) framework. In case of periodic signals, a low-pass filter (ideal or

Butterworth) projects the input signal to a low-pass approximation. Similar for finite signals, the input signal is projected onto a spline space.

In the modulated wideband converter (MWC), first the signal is split into M channels. The random demodulator (RD) and the MWC are based on the compressive sampling (CS) framework, thus the input signal is first modulated by a random sign sequence $p(t)$. In a practical implementation, the source of those is a linear feedback shift register (LFSR). In both cases, the modulator follows a low-pass filter which is an integrator in the RD case and ideal in the MWC case.

The output of the preprocessing stage is the signal $y(t)$ or M signals represented as vector $\mathbf{y}(t)$.

Low-rate sampling The Nyquist rate is denoted as f_N and the corresponding sampling period by T_N . The sampling rate of a single low-rate analog-to-digital converter (ADC) is denoted by f_s and the corresponding sampling period by T_s . The total sampling rate of the sampling stage is given by the sum of all low-rate ADCs. The RD and FRI sampling use only one ADC, therefore $f_{\text{tot}} = f_s$. The MWC has M channels, resulting in $f_{\text{tot}} = M \cdot f_s$. The goal of the sampling stage is to sample with a total sampling rate $f_{\text{tot}} < f_N$.

In contrast to the setup in figure 1.1, the quantizer Q is omitted since quantization is not covered by this thesis.

Reconstruction The main work in sub-Nyquist sampling is done in the reconstruction stage where sampling at a low rate is traded for computational power. Since the underlying signal models (2.10) are non-linear, the reconstruction is non-linear by definition. Because of the three different types of sparsity, this block highly depends on the sampling device.

The RD reconstructs a length- N vector \mathbf{x} which corresponds to the samples of $x(t)$ at the Nyquist rate (distorted by an integrator with cutoff at f_N). This is done by solving a CS problem with dimensions $f_s \times f_N$. The output is directly compared with the Nyquist-rate representation of $x(t)$.

The FRI framework reconstructs K weights and K time delays of a stream of Diracs. This is achieved by matrix operations on the samples, followed by polynomial rooting or a singular value decomposition (SVD). The output is compared with the parameters of the original for the weights and the time shifts separately.

The MWC reconstructs the input signal as vector \mathbf{x} the same way as in the RD case. The output is directly compared with the Nyquist-rate representation of $x(t)$.

Chapter 4

The random demodulator (RD)

The RD is a device which directly applies (discrete) compressive sampling (CS) principles to analog signals. It first was presented in [32]. Apart from ideas of CS, the main working principle is borrowed from telecommunications: Similar to spread spectrum technologies like direct-sequence spread spectrum (DSSS) the input signal first is multiplied by a pseudo-random chipping sequence. The purpose of this multiplication is to spread the information content of a single frequency across the whole spectrum. Afterwards, this sequence is filtered by an analog filter $h(t)$ followed by low-rate sampling at rate f_s . The reconstruction stage recovers the signal $x[n]$ at the Nyquist rate f_N .

4.1 Working principle

4.1.1 The general model

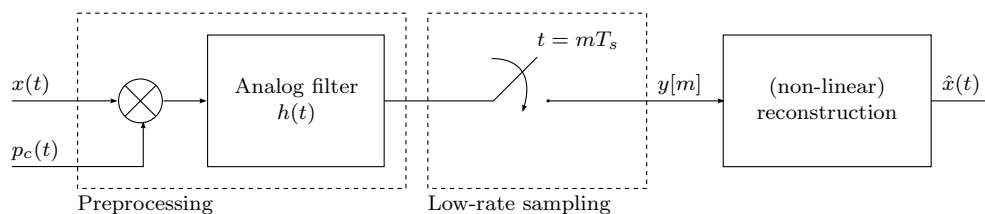


Figure 4.1: The RD: The input signal first is multiplied by a random chipping sequence, afterwards filtered with an analog filter and sampled at a low rate

The block diagram of the RD is shown in figure 4.1. The output signal $y[m]$ of the sampling process can be calculated as [33]:

$$y[m] = \int_{-\infty}^{\infty} x(\tau)p_c(\tau)h(t - \tau) dt \Big|_{t=mT_s} \quad (4.1)$$

It is assumed that the input signal can be represented in a finite dictionary:

$$x(t) = \sum_{n=1}^N c_n \psi_n(t) \quad (4.2)$$

Each of the dictionary entries may have infinite bandwidth. However, the signal $x(t)$ itself has only N degrees of freedom. Using this signal model, (4.1) can be expressed as

$$\begin{aligned} y[m] &= \int_{-\infty}^{\infty} \sum_{n=1}^N c_n \psi_n(\tau) p_c(\tau) h(t - \tau) dt \Big|_{t=mT_s} \\ &= \sum_{n=1}^N c_n \underbrace{\int_{-\infty}^{\infty} \psi_n(\tau) p_c(\tau) h(t - \tau) dt}_{\theta_{m,n}} \Big|_{t=mT_s} \end{aligned} \quad (4.3)$$

which can be expressed as a matrix-/vector multiplication with the $M \times N$ matrix Θ when using M samples:

$$\begin{bmatrix} y[1] \\ y[2] \\ \vdots \\ y[M] \end{bmatrix} = \begin{bmatrix} \theta_{1,1} & \theta_{1,2} & \cdots & \theta_{1,N} \\ \theta_{2,1} & \theta_{2,2} & \cdots & \theta_{2,N} \\ \vdots & \ddots & & \vdots \\ \theta_{M,1} & \theta_{M,2} & \cdots & \theta_{M,N} \end{bmatrix} \cdot \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ \vdots \\ c_N \end{bmatrix} \quad (4.4)$$

or

$$\mathbf{y} = \Theta \cdot \mathbf{c} \quad (4.5)$$

If the sum in (4.2) contains only $K \ll N$ non-zero entries, the signal $x(t)$ is K -sparse with respect to the basis Ψ and therefore \mathbf{s} is a K -sparse vector. If the number of samples $M < N$, (4.5) is an underdetermined set of equations which can be solved using standard tools from the CS framework. The solution vector \mathbf{c} then fully specifies the input signal in (4.2).

4.1.2 Using an integrator as analog filter

Solving the entries $\theta_{m,n}$ in (4.3) analytically is a hard task because $p_c(t)$ is a random sequence. Calculating the entries numerically requires that $\psi_n(t)$ or $h(t)$ have finite (short) support. Therefore the calculation might be difficult in practice. The first

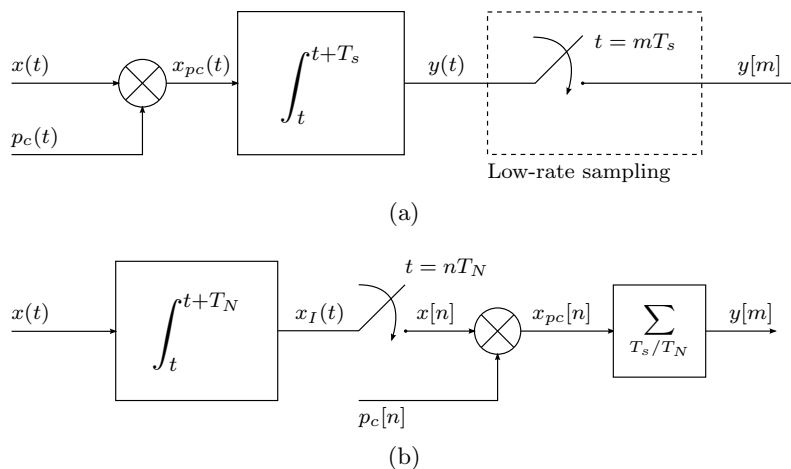


Figure 4.2: (a) Block diagram of the RD when using an integrator. (b) Equivalent diagram, sampling at Nyquist rate. Both versions are equivalent when T_N divides T_s and $p_c(t)$ alternates exactly at the Nyquist rate

practical implementation of the RD was presented in [34]. The block diagram is shown in figure 4.2(a). This version uses an integrator as analog filter. The integrator acts as a low-pass filter which has its cutoff frequency at a low-rate sampling rate f_s . The output samples of the RD coincide with the definition of CS where the sampling process is thought as taking measurements as *inner products* instead of uniform samples:

$$y[m] = \langle x(t), \phi_m(t) \rangle = \int_{-\infty}^{\infty} x(t) \cdot \phi_m(t) dt = \int_t^{t+T_s} x(t) \cdot p_c(t) dt \Big|_{t=mT_s} \quad (4.6)$$

In this architecture the input signal is assumed to be a *discrete multitone* signal which is a special version of (4.2) where the dictionary entries are sinusoids:

$$x(t) = \sum_{n=1}^N c_n e^{-j2\pi f_n t} \quad (4.7)$$

with f_k being the frequencies and c_k their respective amplitudes. Define the set of allowed analog frequencies as Ω . If the signal contains only integral frequencies between $-f_N/2$ and $f_N/2$, this set is defined as:

$$f_n \in \Omega \subset \{0, \pm 1, \pm 2, \dots, \pm(f_N/2 - 1), f_N/2\} \quad (4.8)$$

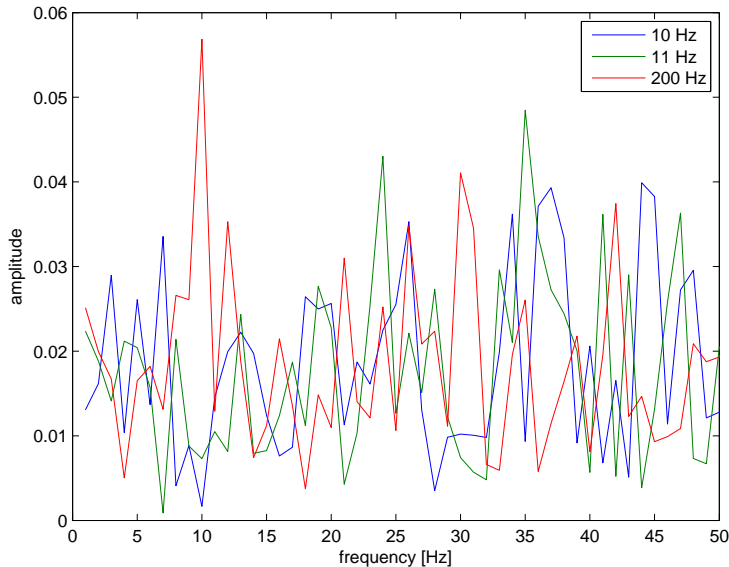


Figure 4.3: Baseband of three different tones: All of the tones have distinct signatures which are nearly orthogonal when the phases are taken into account [34] (code)

with $|\Omega| = N$ and the Nyquist rate of $x(t)$ is f_N . The chipping sequence must alternate between -1 and 1 at (or above) the Nyquist rate f_N :

$$p_c(t) = \alpha_n, \quad t \in [n \cdot T_N, (n + 1) \cdot T_N], \quad \alpha_n \in \{-1, 1\} \quad (4.9)$$

Multiplying $x(t)$ with the random chipping sequence $p_c(t)$ corresponds to a convolution of the spectra $X(f)$ and $P_c(f)$. With the assumption of the model in (4.7), the convolution corresponds to shifted versions of the Fourier transform of the chipping sequence. If there are only $K \ll N$ frequencies active, there is enough information in the frequency range $[0, f_s]$ (baseband of the integrator) to recover the whole signal. Figure 4.3 shows the baseband of the signal $x_{pc}(t)$ for three different tones. Although two of them have almost the same frequency, all of them have distinct signatures. They are nearly orthogonal when the phases are taken into account [34].

The mathematical formulation of the system is best explained by the equivalent system [35] which is shown in figure 4.2(b). This system is equivalent to the system in figure 4.2(a) if the following two requirements are met:

1. T_N divides T_s . In this case it does not matter if the integration is done in analog domain for T_s seconds or if T_s/T_N consecutive pieces of T_N -long integration are summed up in digital domain. This is visualized in figure 4.4.

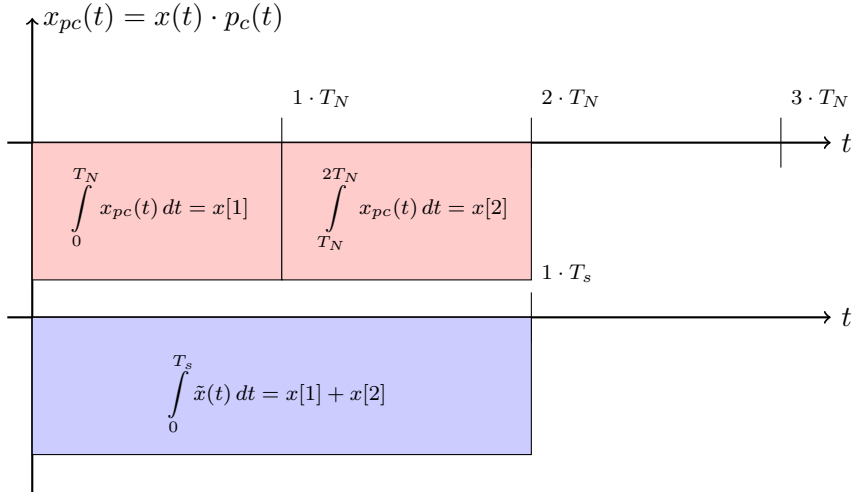


Figure 4.4: Equivalence between figure 4.2(a) and 4.2(b): When T_N divides T_s , T_s/T_N consecutive pieces of T_N -long integration can be summed up in digital domain. In this example: $T_s = 2 \cdot T_N$

2. $p_c(t)$ alternates exactly at the Nyquist rate f_N . In this case $p_c(t) = \text{const.}$ for the T_N -long integration time and therefore this multiplication can be placed after the integration in digital domain as $p_c[n]$.

Following the equivalent system, the signal $x_I(t)$ is obtained by:

$$x_I(t) = \int_t^{t+T_N} x(t) dt \quad (4.10)$$

$$= \sum_{n=1}^N c_n \int_t^{t+T_N} e^{-j2\pi f_n t} dt \quad (4.11)$$

$$= \sum_{n=1}^N c_n \underbrace{\frac{e^{-j2\pi f_n T_N} - 1}{-j2\pi f_n}}_{s_n} \cdot e^{-j2\pi f_n t} \quad (4.12)$$

Since $x(t)$ is assumed to be bandlimited to $f_N/2$ according to the model in (4.7), the samples $x[n]$ directly correspond to $x_I(n \cdot T_N)$:

$$x[n] = x_I(t)|_{t=n \cdot T_N} = \sum_{n=0}^{N-1} s_n e^{-j2\pi f_n n T_N} \quad (4.13)$$

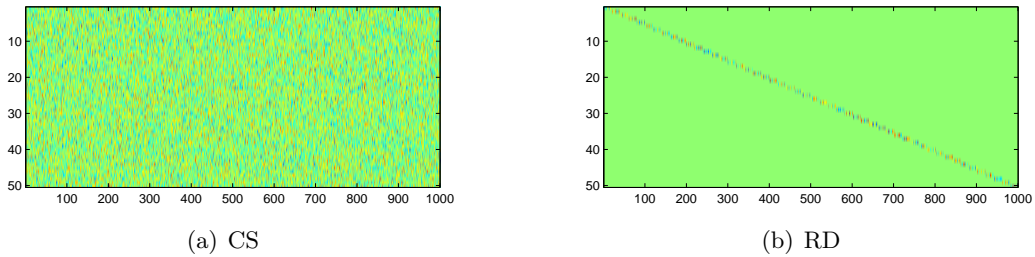


Figure 4.5: Comparison of the sensing matrix Φ between standard CS and the RD _(code)

4.1.3 Derivation of the minimum sampling rate

According to (4.18), this looks like a default CS setup. But as explained in section 2.2, the sensing matrix Φ has to fulfill certain requirements where the restricted isometry property (RIP) is a necessary condition. While it was possible to show that random matrices obey the RIP, it is not so easy to show this for arbitrary matrices. Figure 4.5(a) and 4.5(b) show a typical standard CS sensing matrix next to the $\Phi = \mathbf{H} \cdot \mathbf{A}$ matrix of the RD where the entries have been drawn from a Gaussian distribution. The only random part of the matrix in figure 4.5(b) is the diagonal matrix \mathbf{A} which is scrambled around the main diagonal by the sum and dump matrix \mathbf{H} . Therefore the matrix is sparse and not a typical CS matrix. However, the RD matrix fulfills the RIP except with probability $\mathcal{O}(N^{-1})$ if

$$f_s \geq c \cdot K \log^5(N) \quad (4.20)$$

with c being a specific value [34]. In order to provide a more detailed requirement for the sampling rate, an empirical rule was derived by drawing data vectors and measurement matrices randomly, and solving the system in (4.18) for different sampling rates f_s . In order to guarantee stable recovery the sampling rate should obey the following bound [34]:

$$f_s \geq 1.7K \log(N/K + 1) \quad (4.21)$$

Therefore the RD performs the same way as the standard CS algorithms do: The sampling rate is proportional to the sparsity K and only logarithmically in the Nyquist rate f_N .

4.1.4 Generalization to arbitrary tone spacings

The derivation in section 4.1.2 applies to multitone signals with a tone spacing of 1 Hz and therefore the observation window is 1 s. But it is easy to generalize the signal model

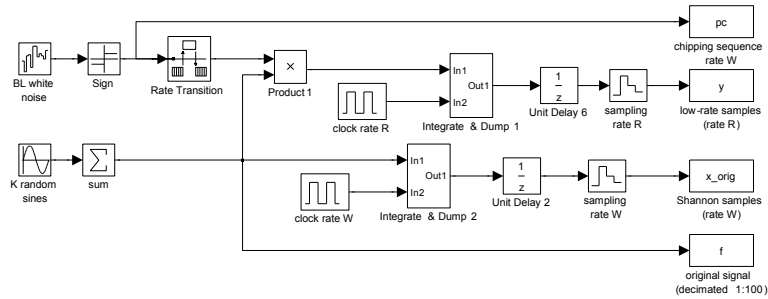


Figure 4.6: SIMULINK model of the RD

in (4.7) to arbitrary tone spacings. When N harmonics are possible with a tone spacing of Δ , the definition of the set Ω in (4.8) becomes [35]:

$$\Omega \subset \{0, \pm\Delta, \pm2\Delta, \dots, \pm(N/2 - 1)\Delta, N\Delta/2\} \quad (4.22)$$

On the one hand, with M samples and the assumption of $f_N = N\Delta$, the sample vector \mathbf{y} has the dimensions $M \times 1$ and the vector \mathbf{x} and \mathbf{s} the dimensions $N \times 1$. Therefore the RD matrix Θ has the dimensions $M \times N$. On the other hand, according to (4.17), the matrix \mathbf{H} has dimensions $M \times MT_s/T_N$ (and therefore also Θ). Consequently, in order to match the dimensions:

$$N = M \frac{T_s}{T_N} = M \frac{f_N}{f_s} = M \frac{N\Delta}{f_s} \quad \rightarrow \quad \Delta = \frac{f_s}{M} \quad (4.23)$$

This means that the tone spacing defines the sampling rate f_s and the number of samples M of the RD and the other way round. In the basic setting $M = f_s$, $\Delta = 1$ and $f_N = N$.

4.2 Implementation

4.2.1 Framework

The RD first was built and analyzed by using the vector-/matrix calculations introduced in section 4.1.2. But this requires that the original signal $x(t)$ and $p_c(t)$ are available exactly at the Nyquist rate. In order to evaluate the RD from a more realistic point of view, an oversampled MATLAB version as well as a SIMULINK model of the RD was built. In general, an oversampling factor from 100 to 1000 was used. Figure 4.6 shows one of the SIMULINK models which were used for the experiments. It implements the model in figure 4.2(a) as well as the equivalent model in figure 4.2(b).

4.2.2 Reconstruction

The reconstruction stage of the RD produces samples of the input signal at the Nyquist rate from the low-rate samples. This is done mainly by solving (4.18) which is a fundamental part of the RD. There is a wide range of solvers available ^{1,2} where most of them are based on convex optimization or greedy algorithms. However, the RD has another requirement: The reconstruction algorithm needs to handle complex values because the output vector \mathbf{s} as well as the sensing matrix Φ is complex. Most of the MATLAB implementations available are not suitable for complex problems. For this reason the same algorithm was implemented which has been used by the authors of the RD: The iteratively reweighted least squares (iRWLS) algorithm [36, 27, 173ff]. A second reason was that the results from [34] should be reproducible. However, this algorithm offers poor performance and is very slow (at least the own implementation). Another option is the CVX toolbox [28] which offers a convenient way to solve optimization problems in MATLAB. For this reason, this toolbox has been used from the beginning.

There is a trick which supports the extension to a broader range of recovery algorithms. A closer look at (4.18) reveals that the only complex part in the RD matrix Θ is the matrix \mathbf{F} which is a permuted discrete Fourier transform (DFT) matrix. This matrix corresponds to the sparsity-inducing matrix Ψ from CS literature which makes the vector \mathbf{x} K -sparse. It is now possible to choose a different basis which has real entries:

$$\mathbf{F}_{\text{real}} = [\Re\{\mathbf{F}_L\} \quad \Im\{\mathbf{F}_R\}] \quad (4.24)$$

where \mathbf{F}_L is the left half and \mathbf{F}_R the right half of the matrix \mathbf{F} [37]. This technique supports to use a broader range of recovery algorithms. Unfortunately the sampling rate f_s has to be doubled in this case because the vector \mathbf{s} is now $2K$ -sparse.

Table 4.1 gives an overview about different reconstruction algorithms and how they work with the RD. They are separated into algorithms which can only deal with a real matrix Θ and algorithms which work also for complex valued problems as required by the RD. The experiments were done by using direct samples $\mathbf{y} = \Theta\mathbf{F}^{-1}\mathbf{x}$, i.e., by solving

$$\min \|\mathbf{s}\|_1 \quad \text{s.t.} \quad \mathbf{F}^{-1}\mathbf{x} = \mathbf{s} \quad (4.25)$$

In all experiments $N = 1000$ and $K = 5$ has been used. When using the real matrix, the most suited algorithm is the orthogonal matching pursuit (OMP) algorithm which is a greedy algorithm. It offers the best signal-to-noise ratio (SNR) and is almost the fastest algorithm. The ordinary matching pursuit (MP) achieves a low SNR: The algorithm recovers the frequency locations exactly, but not the amplitudes. When it

¹<http://sites.google.com/site/igorcarron2/cs#reconstruction>

²<http://dsp.rice.edu/cs>

Algorithm	Sampling rate R	\mathbf{F}	SNR [dB]	Running time [s]
real only				
l1magic ¹	100	real	133.6597	1.1886
l1magic_bpdn ¹	100	real	133.8817	1.2065
lasso ²	100	real	214.6952	1.4776
lars ²	100	real	212.4155	0.83367
omp ²	100	real	260.9824	0.9989
mp ²	100	real	7.8805	4.2988
complex and real				
cosamp ³	50	complex	238.2762	5.7724
cosamp ³	100	real	261.1256	1.6464
cvx (SeDuMi) ⁴	50	complex	137.1817	6.5449
cvx (SeDuMi) ⁴	100	real	211.8597	4.7027
cvx (sdpt3) ⁴	50	complex	130.4159	7.1546
cvx (sdpt3) ⁴	100	real	170.1535	16.8339
irls ⁵	50	complex	24.0526	42.7856
irls ⁵	100	real	128.4567	29.221

¹ ℓ_1 -magic toolbox <http://www.acm.caltech.edu/l1magic/>

² SparseLab <http://sparselab.stanford.edu/>

³ <http://igorcarron.googlepages.com/cosamp.m>

⁴ CVX toolbox [28]

⁵ own implementation

Table 4.1: Performance and running time for different reconstruction algorithms applied to the RD and if they can work with a complex matrix Θ . The CVX implementation offers the best tradeoff and was used during the experiments (code)

comes to algorithms supporting complex problems, the compressive sampling matching pursuit (CoSaMP) [38] algorithm performs the best. However, the algorithm has a serious drawback: The sparsity K needs to be known. The iRWLS algorithm works but it is rather slow and not very reliable. This may be an issue of my own implementation.

The CVX toolbox offers two different solvers where the default solver (self-dual-minimization (SeDuMi)) performs better. Both solvers are able to handle complex problems. The recovery SNR is at the average and knowing the sparsity K is not a requirement. Therefore the best choice among all algorithms is the CVX toolbox with the default solver. As CVX uses optimization techniques the running time is not as good as with the greedy algorithms (MP, OMP, CoSaMP). This is the reason why most of the experiments were made with $N = 1000$ and $K = 5$ (see section 4.3.3).

4.3 Evaluation of the random demodulator

The architecture of the RD seems very simple and straightforward: It supports to implement principles which are known from the CS framework directly. But this approach requires a discretization of the input signal which in turn makes the device operating in discrete domain rather than in analog domain. The device therefore becomes very sensitive to model mismatch.

4.3.1 Signal model

Probably the most fundamental problem of the RD is the restriction to a very small class of signals: Multitone signals which can be described by (4.7). Please note that this signal model requires that $x(t)$ has infinite extend. In this case the spectrum of $x(t)$ has the form

$$X(\omega) = \sum_{n=1}^N c_n \delta(\omega - 2\pi f_n) \quad (4.26)$$

and is therefore sparse. On the other hand, (4.18) requires a discrete signal \mathbf{x} represented by the DFT \mathbf{s} . By contrast, the Fourier transform of analog signals consists of a continuous spectrum and therefore an uncountable number of frequencies. This observation conforms with the derivation in the last section: It requires the equivalence of the figures 4.2(a) and 4.2(b) [35]. In order for the system to work, the input signal is discretized by assumption. Afterwards, a completely discrete system is analyzed and derived.

For the DFT of \mathbf{x} to be sparse, the periods of the frequencies f_n have to be a multiple of the observation window in order to fit the model; otherwise the spectrum is not sparse any more because of the introduced leakage. It corresponds to a sparse signal convolved with the sinc-signal introduced by the observation window. The reconstruction fails in this case because of the slow decay of the sinc-function. If a signal

$$x(t) = c_k e^{-j2\pi f'_k t/N}, \quad f'_k \notin \mathbb{Z} \quad (4.27)$$

is approximated using K harmonic frequencies, the best approximation only satisfies

$$\|x - x_K\|_{L_2[0,1]} \leq K^{-1/2} \quad (4.28)$$

which is a painfully slow rate of decay [34]. Figure 4.7(a) shows that reconstructing multitone signals works very good up to machine precision (with normalized mean squared error (NMSE) of 9e-11). However, reconstruction of $K = 5$ arbitrary frequencies

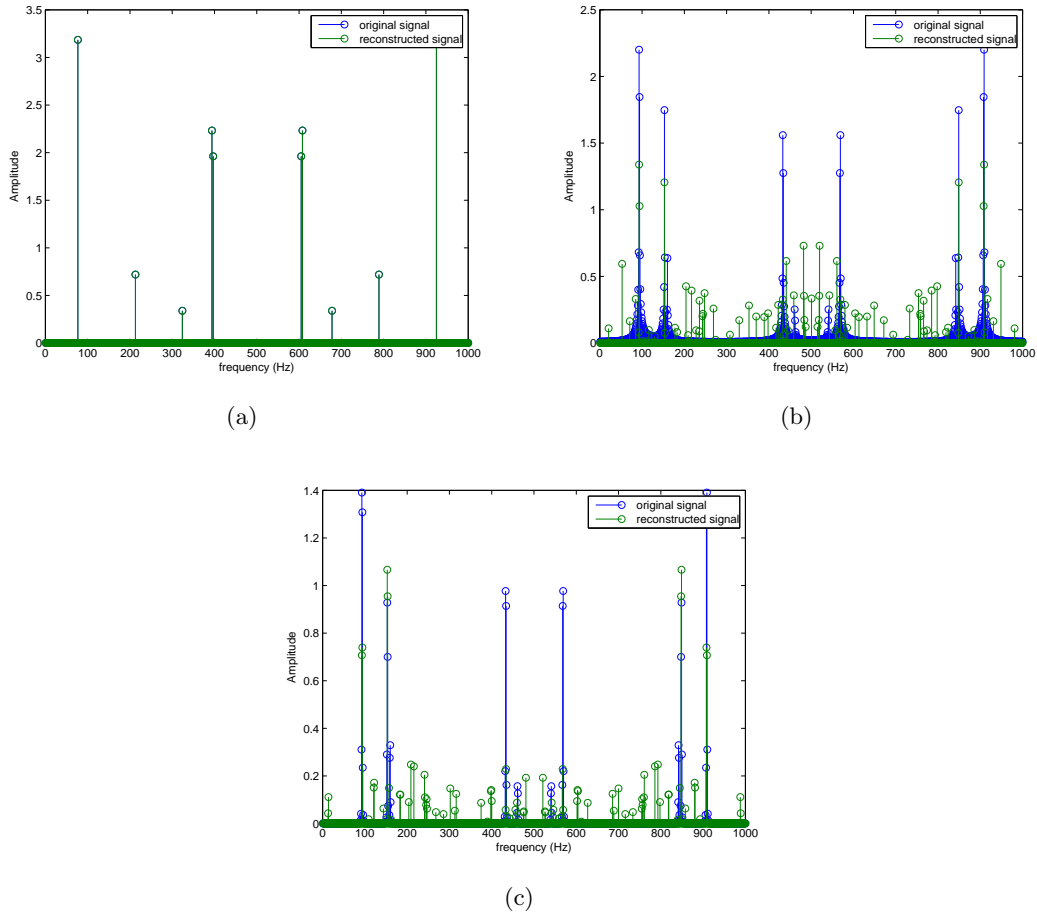


Figure 4.7: Reconstruction of non-multitone signals: (a) multitone signals are reconstructed exactly up to machine precision ($NMSE=9e-11$) (b) reconstruction of arbitrary frequencies fails ($NMSE=1.17$) (c) Hann window applied ($NMSE=1.088$) (code)

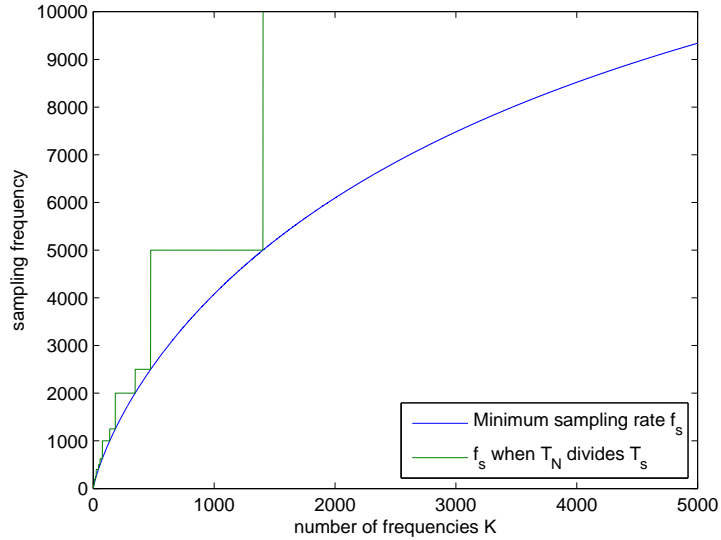


Figure 4.8: Minimum sampling rate versus practical sampling rate: T_N must divide T_s . This requirement considerably lowers the practical achievable sampling rate (code)

completely fails (figure 4.7(b)) with NMSE of 1.17. The performance may be increased by using windowing methods [34] but no concrete details were proposed. Figure 4.7(c) shows reconstruction with a Hann window applied. With an NMSE of 1.088 the performance does not really increase. One option is to increase the sampling rate f_s considerably.

4.3.2 Sampling rate

According to the bound in (4.21) the minimum sampling rate is similar as in the compressive sampling framework. However, it is a necessary condition that T_N divides T_s . In [34] an extension to the matrix (4.17) is proposed which supports two consecutive samples to share contributions from the chipping sequence. For example, with $f_s = 3$ and $f_N = 7$:

$$\mathbf{H} = \begin{bmatrix} 1 & 1 & \sqrt{1/3} & & & & \\ & & & 1 & 1 & \sqrt{1/3} & \\ & & & & & & 1 & 1 & \sqrt{1/3} \end{bmatrix} \quad (4.29)$$

Using this matrix it is not necessary that T_N divides T_s . However, if this is not the case then the equivalence between figure 4.2(a) and 4.2(b) is breached. Therefore in practical implementations the minimum sampling rate f_s according to (4.21) has to be

rounded up to the next integer which divides f_N [35]. Figure 4.8 shows the tradeoff in the sampling rate. For example, when the Nyquist rate is $f_N = 10$ kHz, then the RD samples at the full Nyquist rate if the number of frequencies is greater than $K = 1500$. This shows that the device is only practicable for a small amount of frequencies, in the shown example for $K < 100$.

4.3.3 Reconstruction performance

In [10] Donoho asked:

Can we not just directly measure the part that will not end up being thrown away?

which is a famous quote in CS literature. The RD tries to follow this spirit. However, in contrast to images, which are often directly compressed with compression schemes like JPEG, most electrical signals will be processed by a digital signal processor (DSP) and not only stored in a compressed fashion. Although [39] established first steps towards processing of signals with compressed measurements, usually it will be necessary to reconstruct the uncompressed samples in order to process the data by a DSP. So in some sense the RD could mimic a conventional analog-to-digital converter (ADC) by sampling at a low-rate and afterwards recover the samples at the same rate as a conventional ADC would do. The low-rate ADC could save space or power or could provide a better resolution and therefore a better dynamic range than an ADC which would operate at the Nyquist rate f_N . However, as it can be seen easily from the derivation in section 4.1.2, the complete reconstruction is expected to work at the Nyquist rate, rather than at the information rate K . Even if $x(t)$ would consist only of a few harmonics between 0 Hz and 1 GHz, the whole reconstruction process would restore a 2 GHz signal. This requires a lot of memory and computational resources. The reconstruction, however, is a costly process for two reasons:

1. The dimensions of the CS matrix Θ and the vector \mathbf{s} become very large.
2. The reconstruction process is highly non-linear and requires optimization algorithms which makes the following issues even more relevant.

Memory requirements The most problematic point is the first one [35] which restricts the usage of the RD to low frequency signals. For example, sampling a signal with a bandwidth of 1 GHz at a resolution of 1 Hz would produce an output vector of size $10 \cdot 10^9$. With assumption of $K = 10000$ tones (only 0.001 % active tones) the matrix Θ would have a dimension of $250 \cdot 10^3 \times 1 \cdot 10^9$. This would require a storage of 230 TB at a resolution of only 8 bit. While this amount of storage might be available in future, it should be noted that the reconstruction algorithms also require the whole data to be accessed at least a few times.

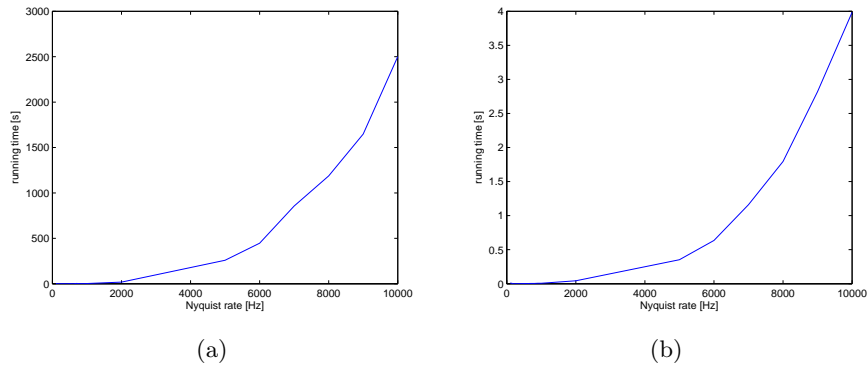


Figure 4.9: Running time of signal recovery of the RD. (a) standard algorithm of CVX optimization toolbox (b) CoSaMP algorithm (`code`)

Computational complexity The authors of the RD argue that the computational cost for the recovery is dominated by the length- N Fourier transform because modern recovery algorithms repetitively apply the system matrix Θ which is a decomposition of constant matrices \mathbf{A} , \mathbf{H} and a DFT matrix \mathbf{F} [34]. But this only applies to greedy algorithms like OMP or CoSaMP. However, even fast algorithms like the CoSaMP algorithm [38] have a running time of $\mathcal{O}(N^2)$ when the sparsity grows linearly with the Nyquist rate. Figure 4.9 shows practical running times of an (unoptimized) MATLAB implementation of an optimization algorithm (CVX toolbox) and the CoSaMP algorithm. For the experiments the sparsity is constant with 0.5% of the Nyquist rate. Although running on a high-speed simulation computer (4 AMD Opteron cores with 2.2 GHz and 16 GB RAM) the practically usable Nyquist rate is below 10 kHz. Most experiments in this thesis were made with a Nyquist rate of 1 kHz, otherwise the reconstruction took too much time. Nevertheless, there may be room for optimizing the reconstruction algorithm.

Real-time processing The RD needs to collect a certain amount of data before it can start the reconstruction process because reconstruction is done at once for a whole observation window. For example, for a signal with $f_N = 1$ kHz and a resolution of 1 Hz the observation window is one sec. The following reconstruction takes additional time. For this reason, the RD is not capable of doing real-time processing. Therefore, the usage of the RD makes only sense in two situations:

1. When the Nyquist rate is very low.
2. When the purpose is data acquisition where the data should be stored rather than being processed.

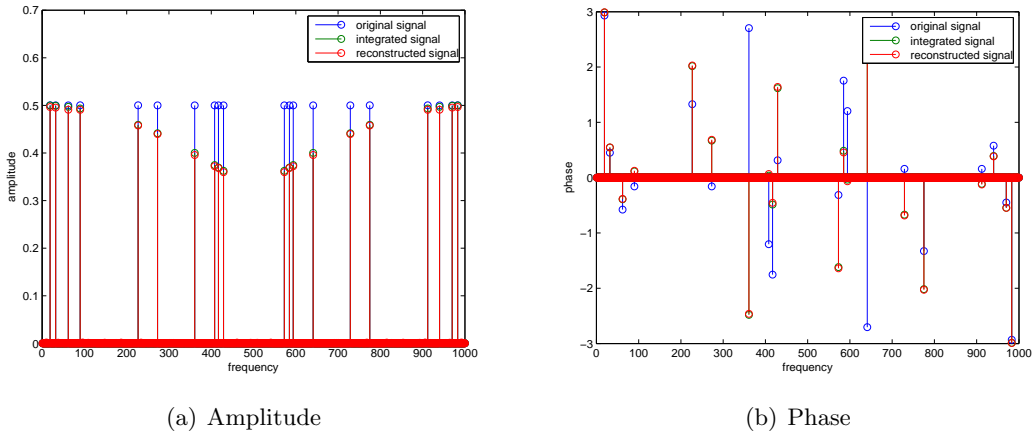


Figure 4.10: Spectrum of original signal $x(t)$, integrated signal $x_I(t)$ and reconstructed signal $y[m]$ (code)

However, especially in the low-rate segment there are cheap, low-power consuming ADCs with a high dynamic range available [5].

4.3.4 The integrator as anti-aliasing filter

The integrator is an essential part of the RD: It supports the equivalence between figure 4.2(a) and 4.2(b). For a theoretical evaluation it should be noted that the reconstructed output $x[n]$ of the RD must be compared with the samples $x_I(nT_N)$ rather than with the samples of the original signal $x(nT_N)$. It is easy to see this by comparing figures 4.2(a) and 4.2(b). This means that the signal is altered in the signal-band even when no frequencies higher than $f_N/2$ are present. Figure 4.10 compares the spectrum for an input signal with $K = 10$ sinusoids with amplitude 1. It is easy to see that the RD recovers the spectrum of the signal $x_I(t)$ precisely whereas amplitude and phase of the signal $x_I(t)$ were altered by the integrator with the transfer function

$$H(s) = \frac{1}{s} (1 - e^{-sT_N}) \tag{4.30}$$

Figure 4.11 shows the amplitude response of such a filter $H(s)$ with a cutoff frequency at f_s compared to a 1st and 2nd order Butterworth filter. The filter has a similar attenuation as a Butterworth filter. The filter attenuates the input signal about T_s of the amplitude over the whole frequency range. This attenuation is implicitly corrected by the recovery algorithm.

Additionally the integrator acts as a low-pass filter in order to reject frequency components higher than f_s . The filter was never addressed by the authors and all experiments

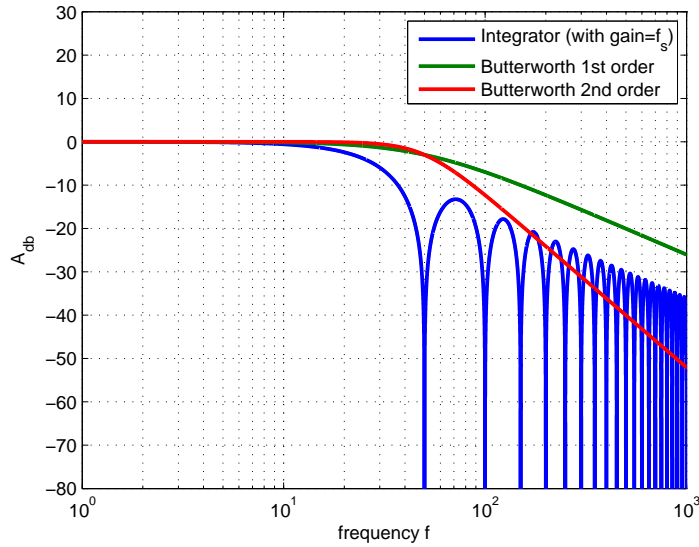


Figure 4.11: Transfer function of the integrator compared to 1st and 2nd order Butterworth filter
(code)

were made under the assumption that the vector \mathbf{s} corresponds to the signal $x_I(t)$ instead of $x(t)$.

4.3.5 Chipping sequence

It is obvious that the chipping sequence must not be shifted because then the random sequence is a different sequence. Since random CS measurements are *democratic* [40, 41] in the sense that each measurement carries the same amount of information it is in principle possible to allow damaged regions of $p_c(t)$.

Sharp transitions

Equations (4.9) and (4.15) require the chipping sequence to be a sign alternating sequence which alternates exactly at the Nyquist rate. If this is not the case, the CS matrix Θ is not constant anymore but will be dependent on the input signal. However, implementation of sharp pulses is not easy in practice because real systems have a certain step response. This is especially true for high-rate signals. Therefore a test with a chipping sequence was made which was resampled from a low-rate version. The sequence was generated using the MATLAB command `resample` and corresponds to low-pass filtering the input signal. Figure 4.12 compares the ideal chipping sequence as required by the equations of the RD versus the interpolated version from a low-rate sequence of signs. The result is

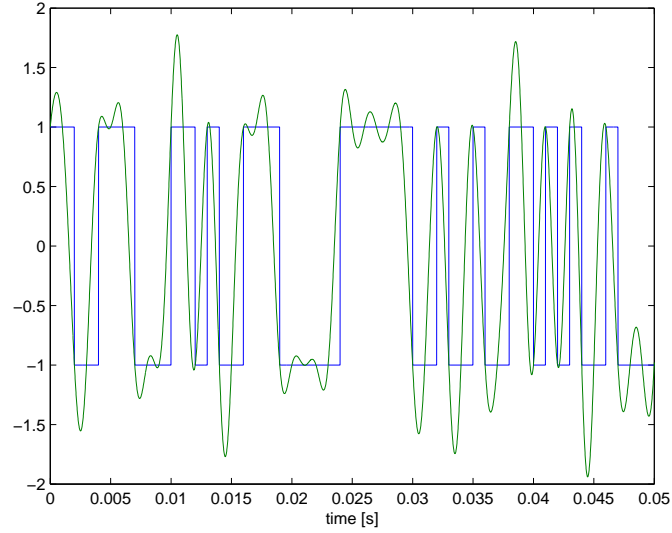


Figure 4.12: Ideal chipping sequence as required by the RD versus low-rate interpolated version (code)

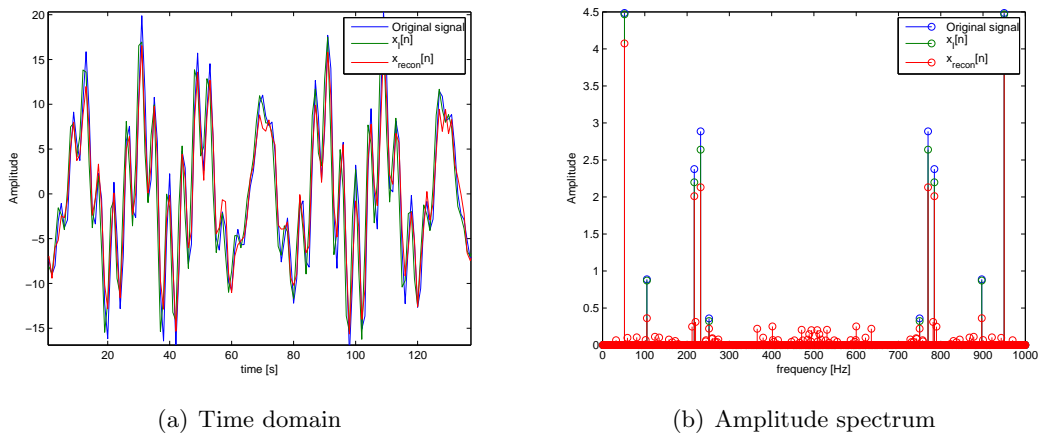


Figure 4.13: Reconstruction using a resampled low-rate signal as chipping sequence instead of the exact alternating version (see figure 4.12). The reconstruction achieves about 10 dB (code)

shown in figure 4.13. The reconstruction achieves an SNR of about 10 dB which keeps the RD working.

In [33], a HSPICE simulation of a prototype circuit was presented which implements the modulator, the integrator and the chipping sequence generator. In the simulation, the implementation worked at a rate of 2 GHz and was able to reconstruct a simple three-tone AM signal with a frequency of 200 MHz. In [42], a hardware prototype was presented which works for 1-3 sinusoids up to 600 kHz. However, no concrete implementation details were given.

Clock jitter

Another experiment examined the sensitivity of the chipping sequence to clock jitter. The clock jitter was modeled in two ways:

- **Periodic jitter:** In this case the sign alternations in (4.9) do not occur exactly at $t = nT_N$ but at $t = nT_N + \Delta t$ where Δt varies periodically according to $\Delta t = A_{UI} \sin(2\pi ft)$. $A_{UI} \in [0, T_N]$ is the jitter amplitude and f the jitter frequency.
- **Random jitter:** In this case, Δt varies according to a normal distribution with variance σ^2 : $\Delta t \sim \mathcal{N}(0, \sigma^2)$.

The results were calculated in terms of the SNR. The following definition has been used:

$$\text{SNR} := 10 \log \left(\frac{\|\mathbf{x}\|_2^2}{\|\mathbf{x} - \mathbf{x}_{\text{recon}}\|_2^2} \right) \quad (4.31)$$

where \mathbf{x} is the vector containing the samples of the original input signal $x_I(t)$ and $\mathbf{x}_{\text{recon}}$ are the reconstructed samples.

Periodic jitter The result of the experiment is shown in figure 4.14. Figure 4.14(a) shows the dependency of the SNR versus the jitter amplitude by fractions of a period of the chipping sequence (UI). The experiment was conducted at different frequencies. The chipping sequence is not very sensitive to the frequency of the jitter but to the amplitude: Already small variations ($< 0.1 \cdot \text{UI}$) decrease the SNR level about two decades. A higher level of jitter prevents the RD from working correctly.

Random jitter Figure 4.14(b) shows the dependency on the SNR in case of random jitter. Already with a variance of 0.1 of the UI the SNR level drops below 10 dB. In general, the RD is very sensitive to a jittered chipping sequence where random jitter has the higher impact.

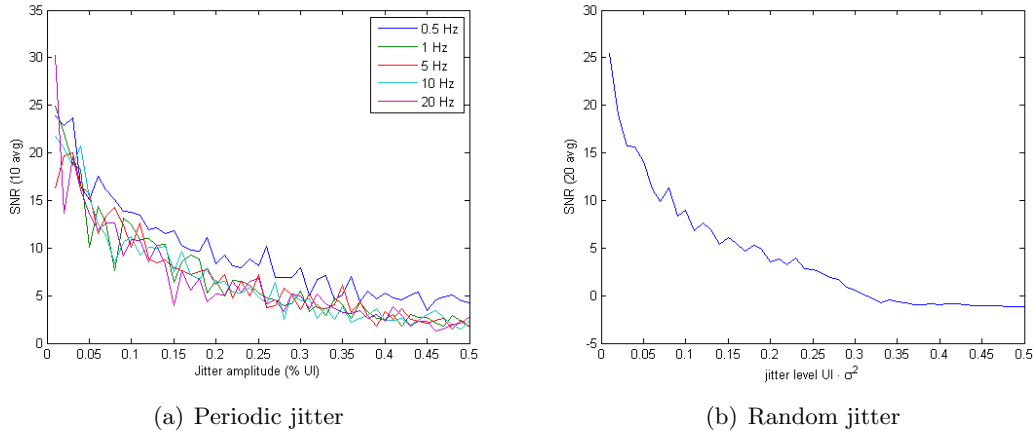
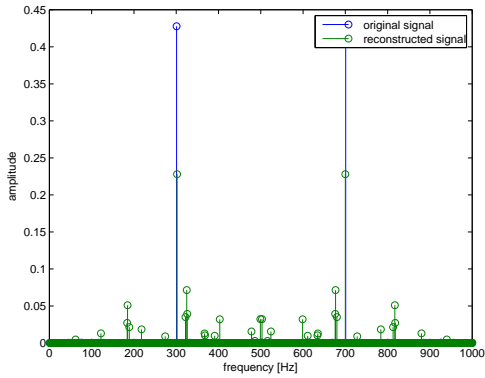


Figure 4.14: Jitter applied to the chipping sequence $p_c(t)$. (a) Periodic jitter at different frequencies. The jitter frequency has few impact $(code)$ (b) Random jitter has a bigger impact on the performance $(code)$

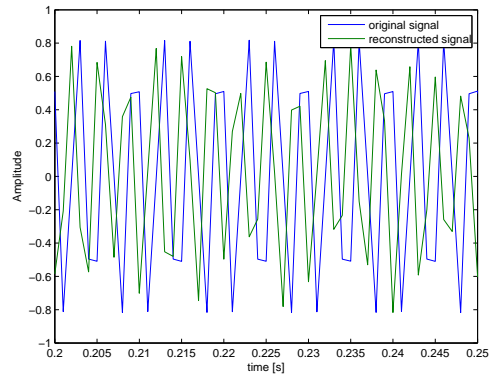
4.3.6 Dependency on the signal model

In [35], it was pointed out that the RD is very sensitive to model mismatch because of the high dependency of the signal model. As explained in section 4.1.4, the tone spacing is related to the sampling rate f_s and the number of samples M . This direct dependency between the signal model and the parameters of the sampling system has the following consequences:

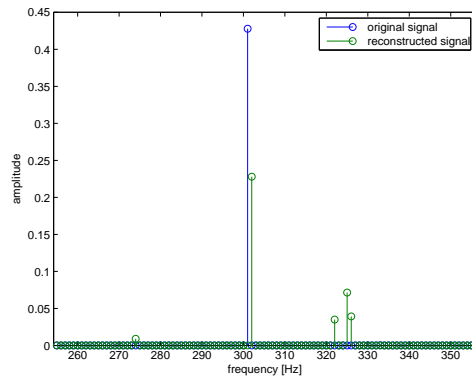
- Given a specific tone spacing Δ , an appropriate RD device has to be built. Only slight changes require a complete redesign of the device.
- If the input signal does not match the grid in (4.22) completely, heavy reconstruction problems occur. Figure 4.15 shows a single sinusoid when there is only a slight drift in the signal model (4.7), (4.22): The tone spacing Δ drifts only about 0.5% from its expected value. The result is that the frequency bin is not matched and spurious tones occur. The signals in the time domain are shifted. The resulting NMSE is 0.0234 which renders the reconstruction useless.
- The device becomes very sensitive to clock jitter. Usually, T_N and T_s will be synchronized by a clock circuit. Synchronization of the clock signals is not easy because it is dependent on thermal noise. Even if the signal matches the signal model exactly, only slight changes in the clock circuit produce similar results as in the previous point.



(a) Amplitude spectrum



(b) Time domain



(c) Zoom into frequency bin

Figure 4.15: Slight drift from the signal model in (4.7) and (4.22) with a single sinusoid: Δ drifts only 0.5% from its expected value. The result is that the reconstruction fails to reconstruct the exact frequency bin and spurious tones occur (code)

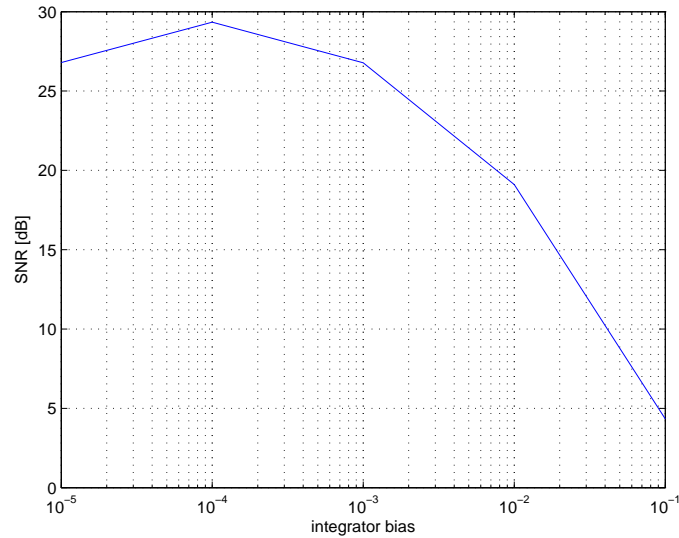


Figure 4.16: Effect on the SNR when the integrator has a bias. The RD reconstructs $K = 5$ multitone signals with amplitude 1 and the measurements are averaged over 30 measurements. When the bias is small, the SNR is dominated by other effects (code)

4.3.7 The integrator

The integrator is another source for model mismatch. When the integrator is not exact, the CS matrix Θ again becomes signal dependent and the correspondence in figure 4.2 is breached. A real integrator is either implemented actively using an operation amplifier, or passively. In both cases a capacitor is the fundamental part which has non-idealities such as serial resistivity and parallel permittivity [35]. As a simple test, a bias was added to the integrator while the RD was reconstructing a sparse multitone signal with amplitude 1 and averaged over 30 trials. The result by means of SNR is shown in figure 4.16. It is shown that a small bias is uncritical. This is shown in the first three values where other effects dominate the SNR.

Furthermore saturation of the integrator is an issue if the integration time takes too long: The input signal must follow a normal distribution around the DC value [42]. An extension [41] describes how to handle saturated measurements by adding a saturation constraint to the recovery algorithm.

4.4 Spectral compressive sampling (SCS)

As discussed in section 4.3, one of the major drawbacks of the RD is the restriction to signals of the form in (4.7). This makes the device almost useless for real systems because the spectrum is only sparse if the tones have either infinite extent or if they are captured such that the observation window is exactly a multiple of the tones' periods. If the signal model is not matched, the frequency bins have a slow rate of decay (see (4.28)), thus making the signal not compressible.

4.4.1 Recovery and sparse approximation algorithm for SCS

One way to overcome this problem is to oversample the DFT, e.g., by using zero padding. The higher the oversampling rate, the higher the compressibility and therefore better recovery performance. Zero-padding of the signal is equivalent to introducing a redundant DFT frame which changes the signal representation with the matrix Ψ . Redundant entries in the matrix Ψ mean higher coherence in the CS matrix Θ and therefore resulting in a worse recovery performance. So there is an inherent drawback in the recovery performance using this method [43]. However, the same paper introduces a new technique called spectral compressive sampling (SCS). The main idea is to change the basis to an oversampled DFT frame $\Psi(c)$ with c being the oversampling factor and at the same time restricting the number of allowed K -sparse signals of length N with techniques known from *model-based compressed sensing* [44, 45]. The result is a structured signal model which restricts the number of those allowed subspaces with the maximum allowed coherence $\mu \in [0, 1]$ between the entries in $\Psi(c)$. First, let (4.13) be rewritten in matrix notation:

$$\mathbf{x} = \sum_{n=1}^N s_n \mathbf{e}(f_n) \quad (4.32)$$

with the vector $\mathbf{e}(f)$ representing the samples of a single complex sinusoid with frequency f :

$$\mathbf{e}(f) = [1 \quad e^{j2\pi f/N} \quad e^{j4\pi f/N} \quad \dots \quad e^{j2\pi f(N-1)/N}]^T \quad (4.33)$$

Using (4.33), the oversampled DFT matrix $\Psi(c)$ may be written as:

$$\Psi(c) = [\mathbf{e}(0) \quad \mathbf{e}(\Delta) \quad \mathbf{e}(2\Delta) \quad \dots \quad \mathbf{e}(N - \Delta)]^T \quad (4.34)$$

where $\Delta = 1/cN$ defines the interval between two consecutive frequencies. The ordinary \mathbf{F} in (4.14) corresponds to $\Psi(1)$ [43]. Now, a signal model $\mathcal{M}_{K,c,\mu}$ can be defined using a

K -sparse version of (4.32) with the oversampled $\Psi(c)$ from (4.34). This model depends on the oversampling factor c of $\Psi(c)$ and the maximum allowed coherence μ :

$$\mathcal{M}_{K,c,\mu} = \left\{ \sum_{k=1}^K s_k \mathbf{e}(d_k \Delta) \quad \text{s.t.} \right. \\ \left. d_k \in \{0, \dots, cN - 1\}, |\langle \mathbf{e}(d_k \Delta), \mathbf{e}(d_l \Delta) \rangle| \leq \mu, 1 \leq k, l \leq K \right\} \quad (4.35)$$

Recovery from measurements \mathbf{y} is performed by an adaption of a model-based iterative hard thresholding (IHT) algorithm. However, any greedy algorithm like CoSaMP will work as well. The difference between the ordinary IHT and the model-based IHT is simple: The best K -term approximation is just replaced with the best K -term model based approximation [44]. In the case of SCS, this is an algorithm which fits a signal \mathbf{x} into the model in (4.35). This sparse approximation algorithm can be implemented by solving a least-squares program and is presented in [43] as well. The resulting recovery algorithm is called spectral iterative hard thresholding (SIHT) via periodogram. The number of required measurements M can be calculated by counting the number of subspaces which define the signal model $\mathcal{M}_{K,c,\mu}$. The resulting number is [43]:

$$M = \mathcal{O} \left(K \log \left(\frac{c(N - K \text{asinc}(1/\mu))}{K} \right) \right) \quad (4.36)$$

Compared to the bound in (4.21) (the sampling rate f_s is directly related to the number of measurements M as explained in section 4.1.4) there is a small reduction of $c \cdot \text{asinc}(1/\mu)$ and an increase by the factor c inside the logarithm. When ignoring the (small) reduction, the number of required measurements corresponds to that of a K -sparse signal under the model (4.7) composed out of cN frequencies.

4.4.2 SCS via spectral estimation

While the algorithm in the last section captures the intuition behind SCS, it suffers from a limited resolution defined by the distance Δ between two frequencies. Instead of using a sparse approximation algorithm for the model in (4.35), the K frequencies can be estimated directly. For this purpose, various spectral estimation methods exist such as multiple signal classification (MUSIC) or estimation of signal parameters via rotational invariant techniques (ESPRIT) [46]. These methods return an estimate of the set of K most dominant frequencies. They do not rely on redundant frames and therefore no control over coherence is required [43]. When using root MUSIC as spectral line estimation algorithm, the new SCS algorithm is called SIHT via root MUSIC. Unfortunately until

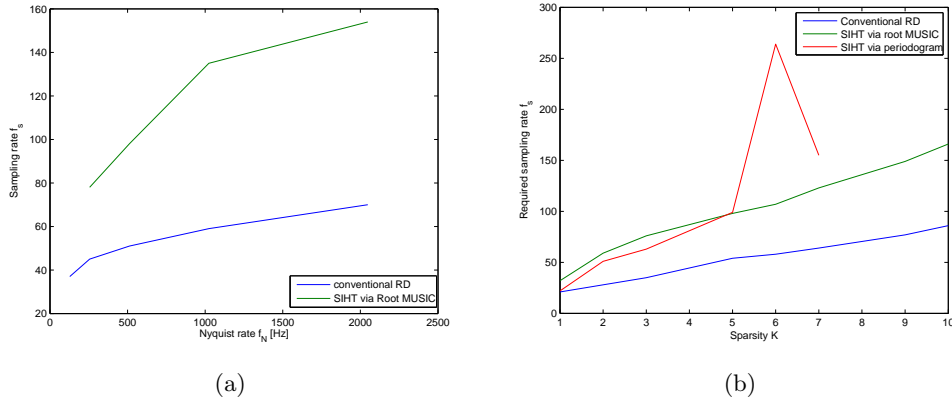


Figure 4.17: Required sampling rate for SIHT: (a) Constant sparsity K , varying Nyquist rate f_N : The sampling requirements are roughly doubled compared to the conventional RD (and even increased fourfold when using a complex matrix \mathbf{F}). SIHT via periodogram is not included since it was not able to reconstruct the signals in most cases (code) (b) constant Nyquist rate $f_N = 500$ Hz any varying sparsity K . Similar as in the first case, the sampling rate needs to be doubled. SIHT via periodogram provides irregular results (code)

now, no derivation for the required number of samples has been found for this case. However, the experiments in the same paper show that this method generally outperforms SIHT via periodogram.

4.4.3 Evaluation of SCS with the random demodulator

The authors of the paper provide a collection of MATLAB functions which implement SIHT via periodogram and SIHT via root MUSIC [37]. These functions have been integrated in the own framework and used for the following experiments.

Reconstruction performance

Figure 4.17 compares the reconstruction performance of SCS with the performance of the conventional RD. Figure 4.17(a) shows the required sampling rate for different Nyquist rates and a fixed sparsity of $K = 5$. Due to the high computational load of the SIHT reconstruction process, the experiments have been averaged over 10 trials only and the set of the Nyquist rates was small ($f_s \in \{128, 256, 512, 1024, 2048\}$). Therefore, the results may not be perfectly accurate. The experiment for a specific sampling rate f_s was declared as success when 9 out of 10 trials reconstructed a signal with more than 20 dB. As discussed in section 4.2.2, it is possible to choose between a real (real matrix \mathbf{F}) and a complex (complex matrix \mathbf{F}) signal representation. In case of a real matrix, the

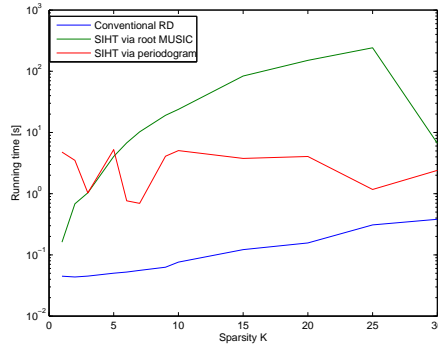


Figure 4.18: Computational load of SIHT: While the conventional RD using the CoSaMP algorithm reconstructs the signals within a few milliseconds, SIHT takes a few seconds and even minutes for the same configuration (`code`)

sparsity needs to be doubled. The SIHT implementation uses a real matrix \mathbf{F} . For the sake of comparability, the experiment for the conventional RD was done by using a real matrix \mathbf{F} as well. The reconstruction algorithm in this case was the CoSaMP algorithm. In case of the conventional RD reconstruction, the K frequencies were multiple integers of the Nyquist rate, whereas in the SIHT case, the frequencies were chosen uniformly from $[0, f_N]$.

As can be seen from figure 4.17, the requirement on the sampling rate is roughly doubled in the SIHT case. If the conventional RD uses a complex matrix \mathbf{F} , it even is four times larger. Apart from this, the diagram lacks the results from SIHT via periodogram. The reason is that SIHT via periodogram is often unable to determine a solution at all.

Figure 4.17(b) shows an experiment with a constant Nyquist rate $f_N = 500$ Hz but varying sparsity $K \in \{1, 2, 3, 5, 6, 7, 9, 10\}$. Due to the high computational load, the experiments were averaged over 10 trials only. It again shows that the required sampling rate is roughly doubled for SIHT via root MUSIC. The requirements for SIHT via periodogram are similar. However, the results are irregular compared to SIHT via root MUSIC. Likewise, reconstruction completely fails with SIHT via periodogram in the case $K = \{9, 10\}$.

In general, SIHT does not provide machine precision reconstruction any more. Furthermore, SIHT via periodogram seems to be very error-prone and rather be usable for theoretical analysis.

Computational load The computational load of SIHT generally is very high. For example, an experiment determining the minimum required sampling rate for a Nyquist rate of 500 Hz and different K takes about one hour on a typical PC for the conventional

RD (using CoSaMP). The same experiment takes about one week with SIHT.

Figure 4.18 shows an example for the running times of the two SIHT algorithms compared to the conventional RD. The experiments used a Nyquist rate of 500 Hz and the sparsity K varied from 1 to 30. For the sake of comparability, the used sampling rates f_s were obtained from the experiment in figure 4.17(b). Specifically, the required sampling rate for the conventional RD was doubled in the case of SIHT via root MUSIC and multiplied by 2.3 in the case of SIHT via periodogram. Therefore the results resemble the running time on a typical PC required to reconstruct K tones in a signal with Nyquist rate $f_N = 500$ Hz. Since the running times are fundamentally different, the results are plotted logarithmically. While the conventional RD recovers the signals in a few milliseconds, SIHT usually requires a few seconds and even minutes.

4.5 Conclusion

The RD is a proof of concept which shows how results from CS can be applied to analog signals. In particular, it introduces an alternative sampling device to conventional ADCs. Probably the most fundamental constraint is the signal model (4.7) which forces equivalence to the DFT by allowing only integral frequencies which in turn assumes a discrete model. The discussion shows that this assumption is breached easily. While this issue can be weakened by SCS to some extent, the RD still has fundamental limitations. Because of the high computational load and memory requirements, the RD as sampling device is only reasonable for low-rate acquisition systems which mainly capture data. A typical application might be power-constrained sensor networks. However, in general the RD shows that principles from CS can not be applied to analog signals directly.

Chapter 5

Sampling at the rate of innovation

The Sampling at finite rate of innovation (FRI) framework is another approach to low-rate sampling below the Nyquist rate. It first was developed in the Ph.D. thesis of Pina Marzillano [13]. In contrast to compressive sampling (CS) approaches like the random demodulator (RD), it is not based on solving a set of underdetermined equations. The framework rather tries to relate parameters of the input signal to the samples analytically by deriving a signal from the samples which contains *signal innovations*. Figure 5.1 shows the sampling setup of the framework. The input signal is first filtered by an (appropriate) analog filter and afterwards sampled at a low rate f_s . The first stage of the reconstruction estimates parameters of the input signal, for example, time delays of pulses. The second, optional decoding stage may reconstruct the input signal in analog domain or provide digital samples at an arbitrary rate. However, the strength of this method is that the reconstruction algorithm directly provides the parameters of the input signal rather than digital samples of the input signal.

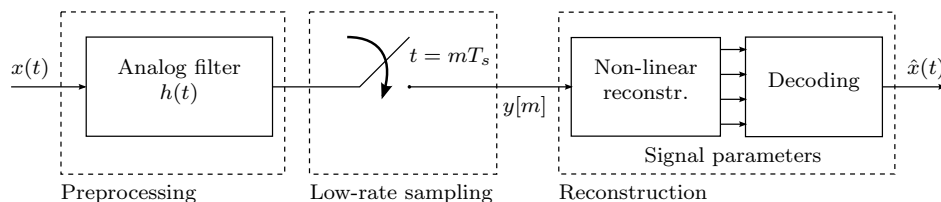


Figure 5.1: General sampling setup of FRI: The input signal first is filtered by an appropriate analog filter and sampled at rate f_s lower than the Nyquist rate. The non-linear reconstruction stage estimates a set of parameters of the signal (e.g., time delays). An optional decoding stage may reproduce the original analog signal

The most important differences compared to the CS framework are:

- The preprocessing and the actual sampling stage are the same as in traditional sampling.
- It does not incorporate any randomness.

- In contrast to the CS framework, it essentially considers *analog* signals rather than discrete approximations or representations.
- The main work is done in the reconstruction stage which highly depends on the signal.
- The output of the non-linear reconstruction stage are parameters of the input signal rather than samples at a specific rate.
- For this reason, the method does not rely on the Whittaker-Kotelnikov-Shannon (WKS) sampling theorem but rather offers a different view on sampling.
- It supports sampling of signals which have theoretically infinite bandwidth.
- The original signal or the samples of the original signal may be reconstructed by using an optional decoding stage.

Until now neither a prototype nor any details for practical implementation have been published and the publications about the FRI framework have a rather theoretic approach. Therefore the FRI framework is fairly theoretic until now.

5.1 Main idea and working principle

The main idea behind the FRI framework is best explained by using a periodic stream of Diracs as input signal. First, suppose the Fourier coefficients $X[m]$ of the input signal are known. Now consider a stream of Diracs with period T where each period consists of K Diracs where the t_k are the time shifts relative to the starting point of the period and the c_k their respective weights:

$$x(t) = \sum_{n \in \mathbb{Z}} \sum_{k=1}^K c_k \delta(t - t_k - nT), \quad 0 < t_k \leq T \quad (5.1)$$

Since this signal is periodic with T it can be described using its Fourier series:

$$x(t) = \sum_{m=-\infty}^{\infty} X[m] e^{j \frac{2\pi}{T} mt} \quad (5.2)$$

The signal in (5.1) has infinite bandwidth, so theoretically it can not be sampled using the WKS sampling theorem. However, the signal itself can be described using only $2K$ parameters: K time delays t_k and K weights c_k . Therefore, the information rate (or the rate of innovation) per period is given by

$$\rho = \frac{2K}{T} \quad (5.3)$$

A very important result of the FRI framework is that the signal $x(t)$ is fully specified by a small set of Fourier coefficients in (5.2). The number of required Fourier coefficients is directly related to the number of Diracs (and therefore to the rate of innovation given by (5.3)) and must contain at least M_F consecutive Fourier coefficients:

$$M_F \geq 2K \quad (5.4)$$

The Fourier coefficients in (5.2) are obtained with:

$$\begin{aligned} X[m] &= \frac{1}{T} \int_0^T x(t) \cdot e^{-j\frac{2\pi}{T}mt} dt \\ &= \frac{1}{T} \sum_{k=1}^K c_k \int_0^T \delta(t - t_k) e^{-j\frac{2\pi}{T}mt} dt \\ &= \frac{1}{T} \sum_{k=1}^K c_k e^{-j\frac{2\pi}{T}mt_k}, \quad m \in \mathbb{Z} \end{aligned} \quad (5.5)$$

Please note that this is an infinite series in general. Now consider some causal filter $A(z)$:

$$A(z) = \sum_{m=0}^K a[m]z^{-m} \quad (5.6)$$

The goal is now to construct the filter $A(z)$ in such a way that the Fourier coefficients in (5.5) cancel out when convolved with the filter $A(z)$:

$$(a * X)[m] = \sum_{i=0}^K a[i]X[m - i] \stackrel{!}{=} 0 \quad (5.7)$$

This filter is not realized physically, that is, no signal is filtered by $A(z)$. It only is used as an analytical tool within the reconstruction process. A filter which satisfies (5.7) is called *annihilating filter* because it annihilates the signal $X[m]$. The annihilation property is fulfilled exactly when $A(z)$ has K zeros at the locations $z_k := e^{-j2\pi t_k/T}$:

$$A(z) = \prod_{k=1}^K (1 - \underbrace{e^{-j2\pi t_k/T}}_{z_k} z^{-1}) \quad (5.8)$$

This can be shown by using the definition of the convolution operator and the definition of (5.5). It follows that [47]:

$$\begin{aligned} (a * X)[m] &= \sum_{i=0}^K a[i]X[m-i] \\ &= \sum_{i=0}^K a[i] \frac{1}{T} \sum_{k=1}^K c_k e^{-j2\pi(m-i)t_k/T} \\ &= \frac{1}{T} \sum_{i=0}^K \sum_{k=1}^K c_k a[i] z_k^{m-i} \\ &= \frac{1}{T} \sum_{k=1}^K c_k z_k^m \underbrace{\sum_{i=0}^K a[i] z_k^{-i}}_{A(z_k)=0 \text{ (cf. (5.8))}} = 0 \end{aligned} \quad (5.9)$$

This means that (5.7) indeed is fulfilled when the filter has the zeros at z_k . On the other hand, a look at (5.8) reveals that the filter $A(z)$ specifies the time delays t_k of the input signal $x(t)$. This is the point: First, obtain enough Fourier coefficients $X[m]$ for solving (5.7). Second, design the filter $A(z)$ with filter coefficients $a[m]$ such that (5.7) is satisfied. Finally, find the roots z_k of this filter in order to express the filter using (5.8). This is the one non-linear step in the reconstruction. Condition (5.7) ensures that the zeros are equal to z_k .

Solving (5.7) requires $2K$ Fourier coefficients ($2K$ values of $X[m]$ need to be accessed in order to guarantee the annihilation requirement). This is the reason for (5.4). For example, choose the set $\Omega = \{-K, \dots, K\}$, then (5.5) becomes:

$$X[m] = \frac{1}{T} \sum_{k=1}^K c_k e^{-j2\pi m t_k/T} = \frac{1}{T} \sum_{k=1}^K c_k z_k^m, \quad m \in \Omega \quad (5.10)$$

Since $X[m]$ and z_k are known in (5.10), this defines a linear system of equations in c_k . Finally, the time delays t_k can be found from the roots z_k with the relation:

$$t_k = \Re \left\{ T \cdot \frac{\log(z_k)}{-j2\pi} \right\} \pmod{T} \quad (5.11)$$

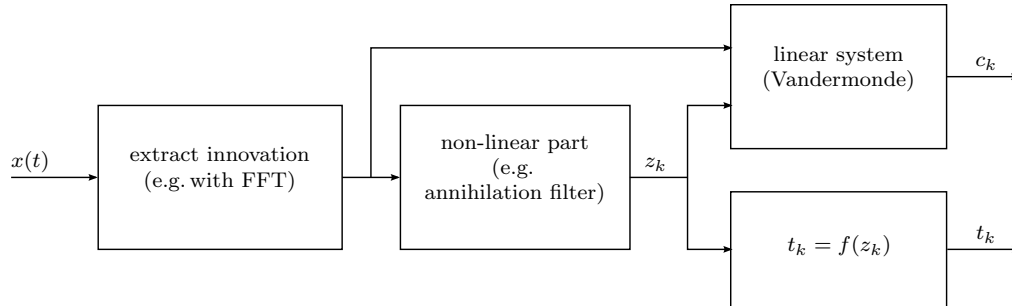


Figure 5.2: Basic principle of FRI reconstruction: Find a signal which encodes *signal innovations*. Afterwards, solve a non-linear problem in order to find the time roots z_k . The time delays t_k are related with the roots z_k by a scalar relation. Solving a linear set of equations finally reveals the weights c_k which together with the time delays t_k describe the original input signal $x(t)$ fully

The K time delays t_k and weights c_k fully specify the input signal $x(t)$ (5.1) now. Figure 5.2 shows a block diagram of the reconstruction. All methods within the FRI framework are based on this concept:

1. Find an *innovation signal* which encodes the signal innovations. Usually this is a sum of exponentials. In the previous example this signal are the Fourier coefficients $X[m]$.
2. From this signal, find the roots z_k by solving a non-linear problem. In the previous example this is done by solving (5.7) followed by polynomial rooting which is the non-linear step.
3. Using the roots z_k , solve a linear system of equations in order to obtain the weights c_k .
4. The time delays t_k are related with the roots z_k by a simple scalar relation, in the previous example (5.11).

5.2 Why Diracs?

In general, the FRI framework deals with signals which can be represented by a finite set of parameters. Unfortunately until now, only derivations for streams of Diracs could be found. The reason for this is that time shifts correspond to modulation in Fourier domain. In the case of Diracs this is nothing more than a sum of sinusoids which can be estimated by the annihilating filter.

One may ask whether it makes sense to design a sampling system for such a specialized class of signals. First, consider a stream of pulses $x'(t)$ with shapes $g(t)$ and with arbitrary

shifts and scales. Such signals occur often in practice, for example in time estimation or ultra-wide band (UWB) (for example [48, 49]). In fact, such a signal corresponds to a stream of Diracs, convolved with the pulse shape $g(t)$. If the pulse shape is known and can be represented in Fourier domain, the Fourier coefficients of a stream of Diracs can be obtained as:

$$X[m] = X'[m]/G[m] \quad (5.12)$$

Second, the authors show that a wide range of signals can be reduced to a stream of Diracs as well. In particular, nonuniform splines and piecewise polynomials can be reduced to a stream of Diracs [31]. However, both methods are based on derivatives which easily lead to numerical problems [50].

This thesis covers only the case of Dirac streams. First, because the other FRI signals may be reduced to a stream of Diracs and second, because the recovery of Dirac streams has to work well in order for the other methods to work.

5.3 The basic case: Periodic stream of Diracs using the sinc kernel

The most basic form of the FRI framework considers a periodic stream of K Diracs and implements the sampling concept as described in section 5.1. It uses the sinc kernel (ideal low-pass filter) as analog filter.

5.3.1 Implementation using the annihilating filter

The remaining question is how the M_F Fourier coefficients can be obtained from the samples $y[m]$. The output of the sampling system shown in figure 5.1 is given by [31]:

$$y[m] = \langle x(t), h(t - mT_s) \rangle = \int_{-\infty}^{\infty} x(t)h(t - mT_s) dt \quad (5.13)$$

Using the signal definition of the Fourier coefficients in (5.2), the samples in (5.13) become:

$$\begin{aligned} y[m] &= \int_{-\infty}^{\infty} \sum_{n \in \mathbb{Z}} X[n] e^{j2\pi nt/T} h(t - mT_s) dt \\ &= \sum_{n \in \mathbb{Z}} X[n] \int_{-\infty}^{\infty} h(t - mT_s) e^{j2\pi nt/T} dt, \quad 2\pi n/T = -\omega \end{aligned}$$

$$\begin{aligned}
 &= \sum_{n \in \mathbb{Z}} X[n] \int_{-\infty}^{\infty} h(t) e^{-j\omega t} dt e^{-j\omega m T_s}, & \text{(shift theorem)} \\
 &= \sum_{n \in \mathbb{Z}} X[n] H\left(\frac{2\pi}{T}n\right) e^{j\frac{2\pi}{T}nmT_s} & (5.14)
 \end{aligned}$$

Equation (5.14) is an infinite sum in general. However, choosing an appropriate filter H only will leave a finite number of components. By choosing the ideal low-pass filter with a bandwidth equal to the rate of innovation (5.3), this sum becomes:

$$y[m] = \sum_{n=-K}^K X[n] e^{j\frac{2\pi}{T}nmT_s} \quad (5.15)$$

Under the assumption of M samples, this yields a system of M equations. When the sampling period T_s is an integer multiple of the signal's period T then the samples are the inverse discrete-time Fourier transform (IDTFT) of $X[n]$ [31]. Therefore, calculating the discrete Fourier transform (DFT) of the samples $y[m]$ provides the required $M = 2K + 1 > M_F = 2K$ Fourier coefficients:

$$X[n] = \sum_{m=0}^{M-1} y[m] e^{-j\frac{2\pi}{T}mnT_s}, \quad -K \leq n \leq K \quad (5.16)$$

From this point of view, now it is clear that the minimum number of required samples must be at least $M = 2K + 1$ which is one sample larger than required by (5.4). The required sampling rate is thus given by

$$f_s = \frac{M}{T} = \frac{2K + 1}{T} \quad (5.17)$$

Until now, the bandwidth of the analog filter was exactly the same as the rate of innovation given by (5.3). In this case, the sampling rate is $f_s = (2K + 1)/T$. Although only $2K$ samples are required by the annihilating filter, the analog filter $h(t)$ may have an arbitrary bandwidth, as long as the bandwidth is higher than the rate of innovation given by (5.3) [31]

$$B \geq \rho \quad (5.18)$$

In this case, the spectrum includes all frequency components in the baseband B of the filter $h(t)$ such that the sum in (5.15) becomes ($M_0 > K$):

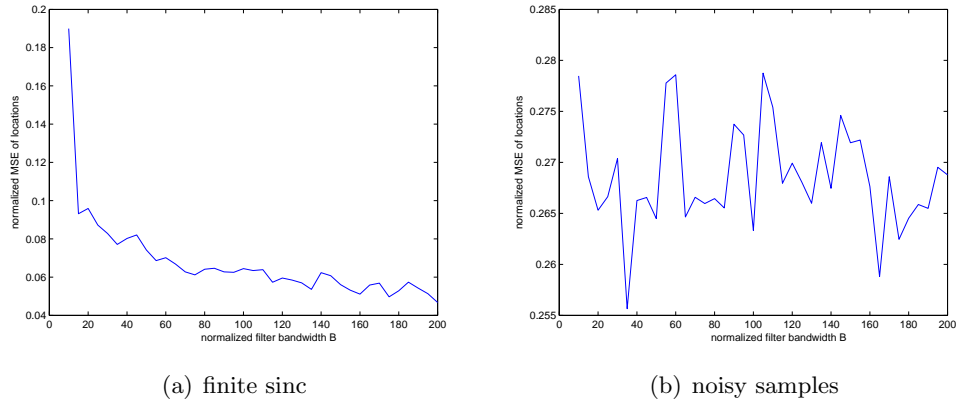


Figure 5.3: Reconstruction performance when increasing the bandwidth B (a) In this case the response of the sinc-filter was cut such that the length was $5T$, yielding an imperfect filter. As the filter bandwidth increases, the recovery performance becomes better (code) (b) Recovery using a perfect sinc filter but using a noisy signal. The recovery performance does not increase when increasing the filter bandwidth (code)

$$y[m] = \sum_{n=-M_0}^{M_0} X[n] e^{j\frac{2\pi}{T}nmT_s} \quad (5.19)$$

and because M_0 must be an integer:

$$M_0 = \left\lfloor \frac{BT}{2} \right\rfloor \quad (5.20)$$

The number of Fourier coefficients and therefore the number of samples M now is given by:

$$M = 2M_0 + 1 = 2 \left\lfloor \frac{BT}{2} \right\rfloor + 1 \quad (5.21)$$

According to (5.17) this also automatically increases the sampling rate. The bandwidth B can be used as a tuning parameter and forms the basis for more advanced recovery strategies. Although only $M_F = 2K$ Fourier coefficients are being used by the annihilating filter, the “quality” of the coefficients becomes better as the bandwidth of the filter increases. The reason is (5.21): The higher the bandwidth, the higher the number of samples. According to (5.16), each coefficient contains information from more samples

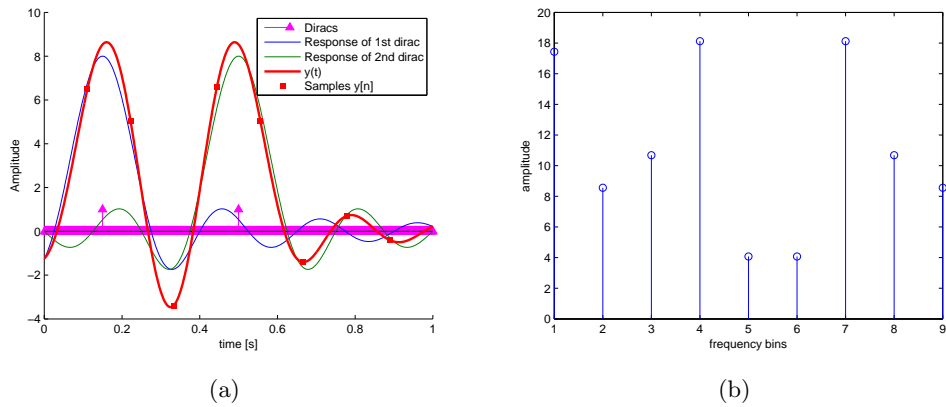


Figure 5.4: (a) Sampling a signal consisting of two Diracs: The filter output is a sum of shifted sinc functions and is sampled at a low rate, in this case $f_s = 9/T$. (b) DFT of the samples $y[n]$. This *innovation signal* contains enough information for recovering the time delays and amplitudes, e.g., with the annihilating filter method `(code)`

than necessary. Increasing the bandwidth provides better recovery performance when using non-ideal filters. However, it does not help when using noisy samples. This is shown in figure 5.3: It shows the recovery performance of the locations t_k using a finite sinc filter of length $5T$. Increasing B provides better results when estimating the Dirac locations. An easy interpretation is: A higher bandwidth means a compacter extension in time. Therefore the responses are more compact and more side lobes of the sinc function are contained in the filter. Figure 5.3(b) shows the same diagram, except an ideal sinc filter was used and 20 dB of noise was added to the samples. Increasing B does not provide better results. The reason is that (5.7) still only uses $2K$ Fourier coefficients.

Figure 5.4 shows an example of sampling two Diracs with amplitude one for one period with $T = 1$. The bandwidth of the filter was chosen to be $B = 2\rho = 2\frac{2K}{T} = 8$. Each Dirac has a shifted sinc-function as response. The sum of the responses for all Diracs corresponds to the output $y(t)$ of the analog filter. According to (5.21), the number of samples is 9. The red squares show those samples. Figure 5.4(b) shows the obtained Fourier coefficients $X[m]$ which have been calculated using the DFT from the samples $y[m]$. From this *innovation signal*, the time delays are estimated.

The second question is how to solve (5.7). This can be written in matrix form:

$$\underbrace{-a[0] \begin{bmatrix} X[1] \\ X[2] \\ \vdots \\ X[K] \end{bmatrix}}_{\mathbf{w}_1} = \underbrace{\begin{bmatrix} X[0] & X[-1] & \cdots & X[1-K] \\ X[1] & X[0] & \cdots & X[2-K] \\ \vdots & \vdots & \ddots & \vdots \\ X[K-1] & X[K-2] & \cdots & X[0] \end{bmatrix}}_{\mathbf{W}_2} \cdot \underbrace{\begin{bmatrix} a[1] \\ a[2] \\ \vdots \\ a[K] \end{bmatrix}}_{\mathbf{a}'} \quad (5.22)$$

The matrix \mathbf{W}_2 is a square Toeplitz matrix with dimensions $K \times K$. When setting $a[0] = 1$, the coefficients of the annihilating filter $a[m]$ for $m = 1, \dots, K$ can be obtained by solving a linear system:

$$\mathbf{a}' = \mathbf{W}_2^{-1} \cdot \mathbf{w}_1 \quad (5.23)$$

so that the annihilating filter coefficients are given by

$$\mathbf{a} = [1 \quad \mathbf{a}']^T \quad (5.24)$$

From the vector \mathbf{a} the roots of the filter can be obtained by polynomial rooting, in order to obtain the K zeros z_k of (5.8). This is the only non-linear step in the reconstruction stage. Afterwards, the K time delays are calculated using (5.11). The remaining K amplitudes c_k are calculated by using the z_k in (5.10):

$$X[m] = \frac{1}{T} \sum_{k=1}^K c_k z_k^m \quad (5.25)$$

This can be written again in matrix form:

$$\underbrace{T \cdot \begin{bmatrix} X[0] \\ X[1] \\ X[2] \\ \vdots \\ X[K-1] \end{bmatrix}}_{\mathbf{v}_1} = \underbrace{\begin{bmatrix} 1 & 1 & \cdots & 1 \\ z_1 & z_2 & \cdots & z_K \\ z_1^2 & z_2^2 & \cdots & z_K^2 \\ \vdots & \vdots & \ddots & \vdots \\ z_1^{K-1} & z_2^{K-1} & \cdots & z_K^{K-1} \end{bmatrix}}_{\mathbf{V}_2} \cdot \underbrace{\begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ \vdots \\ c_K \end{bmatrix}}_{\mathbf{c}} \quad (5.26)$$

The matrix \mathbf{V}_2 is a Vandermonde matrix which has full rank when the Dirac locations are distinct. Therefore the unique solution can be calculated by

$$\mathbf{c} = \mathbf{V}_2^{-1} \cdot \mathbf{v}_1 \quad (5.27)$$

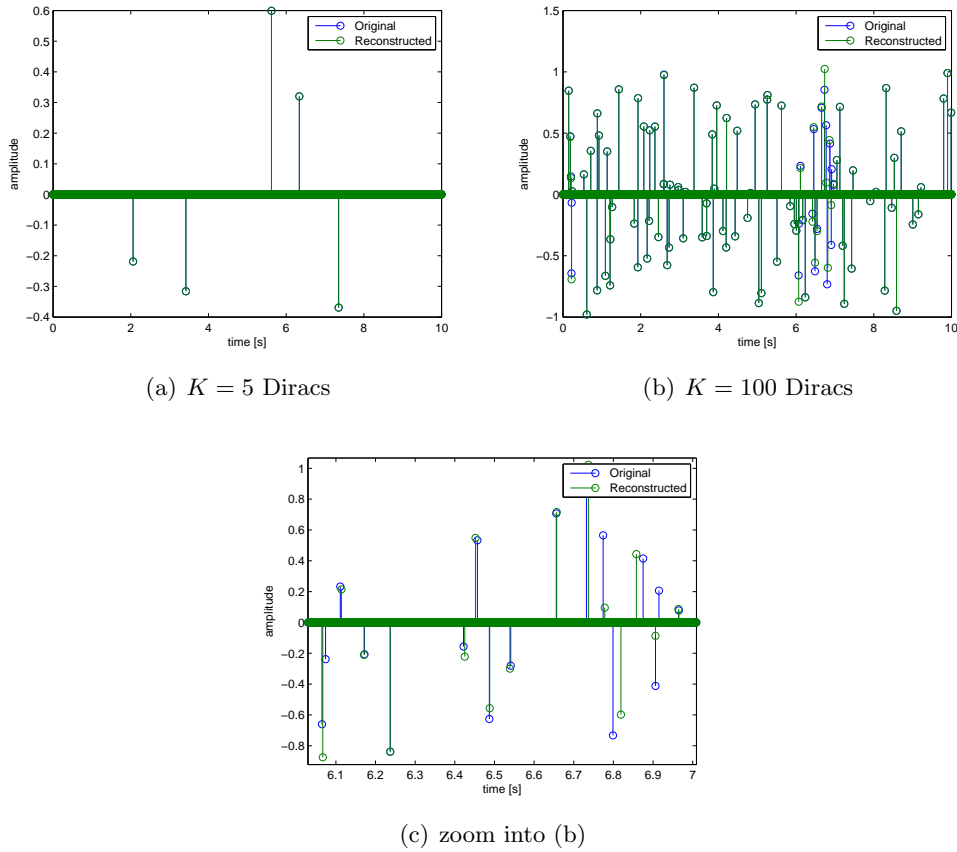


Figure 5.5: Stability of the reconstruction process when the number of Diracs increases (a) a small number of Diracs is always reconstructed up to machine precision (b) With a high number of Diracs most of them are reconstructed successfully but some amplitudes might not be exact and there are a few spurious Diracs (c) Zoom into a “problematic” area: Some Dirac locations have not been reconstructed exactly but very precisely (code)

5.3.2 Stability of recovery algorithm

An important question is if the reconstruction system is stable when the number of Diracs K increases. Figure 5.5 gives a qualitative answer to this question. Figure 5.5(a) shows the original signal and the reconstructed signal. The reconstruction is exact up to machine precision. Figure 5.5(b) shows a more intricate example with $K = 100$ Diracs. Most of the Diracs are perfectly reconstructed, some of them have wrong amplitudes and very few are spurious.

5.3.3 Error measures

Before continuing with a quantitative evaluation of the stability it is important to define an error measure. Until now, the signal-to-noise ratio (SNR) has been used as an error measure which is defined as in (4.31) in the previous chapter. However, this definition is problematic in the case of Diracs as can be understood easily by looking at figure 5.5(c). It shows a zoomed version of figure 5.5(b). Although the time delays have been reconstructed relatively well, some of them do not match exactly. Calculating the error of such a signal yields a huge error; in fact the absolute error is almost doubled although the amplitudes might have been reconstructed very precisely. For this reason it makes sense to calculate the error of the locations and the amplitudes separately. Define the normalized mean squared error (NMSE) as:

$$\text{NMSE}_{\text{loc}} := \frac{\sum_{k=1}^K |t_k - t_{k,\text{recon}}|^2}{\sum_{k=1}^K |t_k|^2} \quad (5.28)$$

$$\text{NMSE}_{\text{val}} := \frac{\sum_{k=1}^K |c_k - c_{k,\text{recon}}|^2}{\sum_{k=1}^K |c_k|^2} = \frac{\|\mathbf{c} - \mathbf{c}_{\text{recon}}\|_2^2}{\|\mathbf{c}\|_2^2} \quad (5.29)$$

In order to define a single error measure it would make sense to compare the Euclidean distance of the Diracs to the origin. This can be done by regarding pairs of (t_k, c_k) as complex numbers:

$$\text{NMSE}_{\text{both}} := \frac{\sum_{k=1}^K |(t_k + jc_k) - (t_{k,\text{recon}} + jc_{k,\text{recon}})|^2}{\sum_{k=1}^K |t_k + jc_k|^2} \quad (5.30)$$

Figure 5.6(a) shows the error of the Dirac locations according to (5.28) as the number of Diracs K increases. After an initial increase of the error, the NMSE keeps at a constant level as the number of Diracs increases. This promises a stable reconstruction process even with a high number of Diracs.

The direct usage of (5.27) for reconstructing the amplitudes produces an extraordinary high error. The reason is shown in figure 5.6(d). It shows the reconstruction of $K = 100$

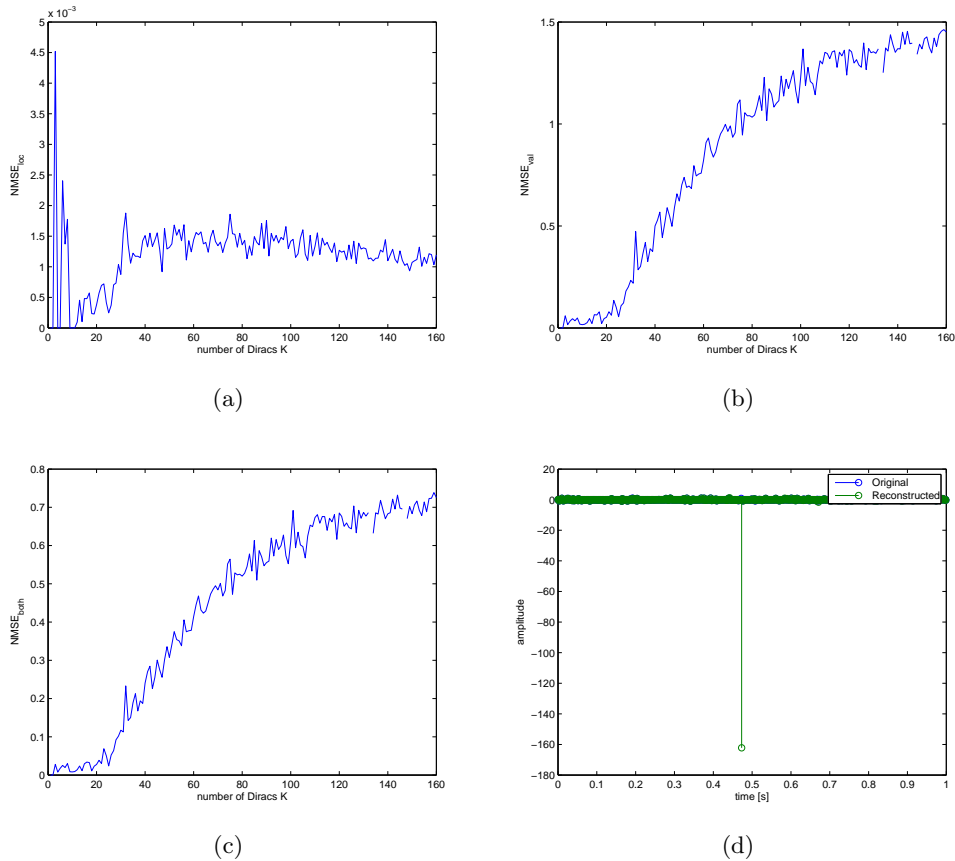


Figure 5.6: Stability of the recovery using the annihilating filter (a) The reconstruction of the Dirac locations works quite well: As the number of Diracs increases, the NMSE keeps nearly constant (code) (b) The reconstruction of the amplitudes is error-prone: As the number of Diracs increases, the NMSE of the amplitudes becomes larger (code) (c) The NMSE as defined by (5.30) is dominated by the worse recovery of the amplitudes and therefore offers limited usage (code) (d) When using (5.27) directly to solve for the amplitudes, the error becomes very huge. In this example, a single Dirac has an amplitude which is 160 times larger than the original Dirac

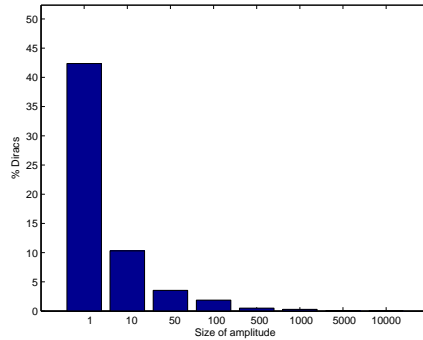


Figure 5.7: Number of failing amplitude recoveries input signals with $K = 100$ Diracs and a maximum amplitude of 1. Using 1000 independent trials, in 40% of the trials, at least one amplitude is above 1. In 2.5% of the trials at least one amplitude is above 100 (code)

Diracs. Although almost all locations and amplitudes could be reconstructed successfully, there is a single amplitude which is 160 times larger than the remaining signal and therefore, it produces a huge error. Figure 5.7 shows an experiment reconstructing $K = 100$ Diracs and measuring the largest peak. The amplitudes of the original signal are distributed equally between $[-1, 1]$. Already 40% of the trials reconstructed an amplitude larger than 1, 2% reconstructed an amplitude larger than 100. Therefore reconstructing the amplitudes according to (5.27) seems to be very error-prone. The reason is that the matrix \mathbf{V}_2 in (5.27) is bad conditioned due to the Vandermonde structure. In order to improve the results, a constraint was added such that the amplitudes do not exceed a specific value, in this case one. Using (5.26) the vector \mathbf{c} was sought such that:

$$\mathbf{v}_1 = \mathbf{V}_2 \cdot \mathbf{c} \quad \text{s.t.} \quad |c_k| \leq 1, \quad \forall k \quad (5.31)$$

Using this strategy, the reconstruction performs better. Figure 5.6(b) shows the error of the Dirac amplitudes. However, as the number K of Diracs increases, the NMSE increases as well. The flattening is due to the constraint $|c_k| \leq 1$. In any case, there is room for improving this issue.

Figure 5.6(c) shows the error according to (5.30). It is dominated by the huge error of the amplitude reconstruction. Therefore (5.30) has limited usage as an error measure. Furthermore, the framework is more suited for problems where time shifts need to be estimated rather than exact amplitude values.

5.4 Other recovery strategies

The reconstruction using the annihilating filter presented so far has the following problems:

1. Equation (5.7) exactly needs $2K$ Fourier coefficients. It is not possible to provide the recovery process with more samples.
2. The recovery requires the exact knowledge of the number of Diracs K .
3. The annihilating filter offers poor noise robustness [51, 47] because in the presence of noise, (5.7) is not exactly fulfilled.
4. Reconstruction of the amplitudes is error-prone.

The authors proposed various alternative reconstruction strategies. Equation (5.10) is nothing more than a sum of exponentials (and therefore sinusoids). That is, finding the frequencies corresponds to spectral estimation. For example, the FRI toolbox [52] implements the *power method* by estimating the power peak in the spectrum which works very reliable even in the presence of high noise. Unfortunately this method only works for a single Dirac. However, any spectrum estimation method like Pisarenko's method, multiple signal classification (MUSIC) or estimation of signal parameters via rotational invariant techniques (ESPRIT) [46] is a good candidate for recovering FRI signals.

5.4.1 SVD approach

As a first step, the annihilating filter method can be extended by using a longer filter [47]: a closer look at the derivation in (5.9) implies that the filter $A(z)$ may be longer: The annihilation property is satisfied as long as all z_k are zeros of $A(z)$. Define $L \geq K$ the maximum number of allowed Diracs, then (5.7) becomes

$$\sum_{i=0}^L a[i]X[m-i] = 0 \quad (5.32)$$

which can again be written in matrix form as [47]:

$$\underbrace{\begin{bmatrix} X[-M_0+L] & X[-M_0+L-1] & \cdots & X[-M_0] \\ X[-M_0+L+1] & X[-M_0+L] & \cdots & X[-M_0+1] \\ \vdots & \vdots & \ddots & \vdots \\ X[M_0] & X[M_0-1] & \cdots & X[M_0-L] \end{bmatrix}}_{\mathbf{w}} \cdot \underbrace{\begin{bmatrix} a[0] \\ a[1] \\ \vdots \\ a[L] \end{bmatrix}}_{\mathbf{a}} = 0 \quad (5.33)$$

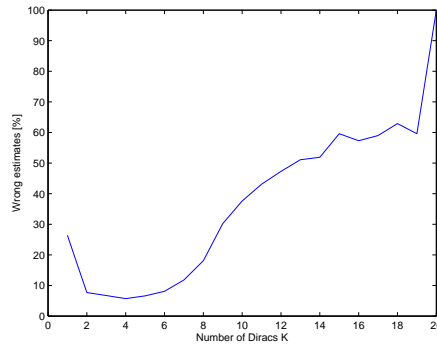


Figure 5.8: Estimating the number of Diracs K when $K < L = 20$ from (5.33): Estimation works well for small values of K . However, already for $K = 13$ the probability of a wrong estimate is larger. Therefore this method is not reliable in general (code)

The matrix \mathbf{W} again is a Toeplitz matrix with dimensions $(M - L) \times (L + 1)$. With $L > K$ there are $L - K + 1$ independent polynomials of degree L . Therefore, the rank does not exceed K . This observation provides a first way to determine K if it is not known a priori: For $L' = 1, \dots, L - 1$, determine the smallest $(M - L') \times (L' + 1)$ matrix \mathbf{W} which is singular. Then $K = L' - 1$ [47]. However, this method is not very reliable. Figure 5.8 shows an experiment estimating different input signals using an FRI sampling system with $L = 20$. The number of Diracs K was increased from 1 to 20 Diracs. The following condition was used as a measure for the singularity of the matrix \mathbf{W} :

$$\mathbf{W} \text{ is singular} \quad \Leftrightarrow \quad |\det(\mathbf{W}^H \mathbf{W})| < \epsilon \quad (5.34)$$

The experiment was conducted 1000 times. Figure 5.8 shows that estimation works when $K \ll L$ but already at $K = 13$, the probability of wrong estimation is larger than the probability of successful estimation.

As mentioned before, in the presence of noise the annihilation equation and therefore (5.33) is not fulfilled exactly any more. However, the minimization $\|\mathbf{W} \cdot \mathbf{a}\|^2$ under the constraint $\|\mathbf{a}\|^2 = 1$ yields a good estimate of \mathbf{a} . This task can be achieved by performing a singular value decomposition (SVD) [53] on the matrix \mathbf{W} : $\mathbf{U}\mathbf{S}\mathbf{V}^T = \mathbf{W}$. The last column of the matrix \mathbf{V} provides the annihilating filter coefficients [47]. This *SVD approach* makes it possible to incorporate more than $2K$ Fourier coefficients to the reconstruction process and thus achieves a better noise robustness when using more samples. On the other hand, the computational cost is more expensive as it requires performing an SVD on a matrix whose size depends on the number of samples M . Furthermore, the authors propose an additional denoising step termed *Cadzow denoising* [54] which is based on iteratively approximating a Toeplitz matrix by averaging over main diagonals [47].

5.4.2 Subspace approach

Another method termed *subspace approach* was presented in [51]. It was shown that the filter length should be related to the size of the data set, rather than to the number of signal components [50]. This fact imposes high requirements on the reconstruction algorithm as it requires performing the SVD on large matrices and afterwards obtain the time delays by polynomial rooting. The subspace approach directly obtains the roots z_k from a matrix decomposition. First, an $M' \times N'$ Hankel matrix \mathbf{W} is built from the $X[m]$:

$$\mathbf{W} = \begin{bmatrix} X[-M_0] & X[-M_0 + 1] & \cdots & X[-M_0 + N' - 1] \\ X[-M_0 + 1] & X[-M_0 + 2] & \cdots & X[-M_0 + N'] \\ \vdots & \vdots & \ddots & \vdots \\ X[-M_0 + M' - 1] & X[-M_0 + M'] & \cdots & X[M_0] \end{bmatrix} \quad (5.35)$$

where $M', N' > K$ and $N' = M - M'$. Second, an SVD is performed: $\mathbf{W} = \mathbf{U} \cdot \mathbf{S} \cdot \mathbf{V}^H$. Now the roots z_k are obtained as the eigenvalues of the matrix \mathbf{Z} :

$$\mathbf{Z} = \mathbf{U}_{S_1}^\# \cdot \mathbf{U}_{S_2} \quad (5.36)$$

The matrix \mathbf{U}_{S_1} contains the first K columns of \mathbf{U} and has the last row omitted. The matrix \mathbf{U}_{S_2} is the same as \mathbf{U}_{S_1} but has the first row omitted instead.

5.4.3 Evaluation

Now it is time to evaluate the stability of the SVD method and the subspace approach compared to the annihilating filter method.

Stability Figure 5.9 shows the reconstruction stability for the locations as well as for the amplitudes as K increases. The experiments were conducted at critical sampling ($B = \rho$). Figure 5.9(a) shows the location NMSE. The SVD, as well as the subspace approach have a nearly constant error as K increases. They even have a slightly better performance than the annihilating filter method. However, the reconstruction of the amplitudes via (5.27) does not improve as can be seen in figure 5.9(b). The reason is that both, the SVD and the subspace approach, improve the recovery of the roots z_k . However, the Vandermonde system in (5.27) keeps the same.

Noise robustness Figure 5.10 shows the NMSE for different noise levels. In this experiment, the samples $y[n]$ for $K = 5$ Diracs have been buried in white noise in order to yield a specific noise level. The experiment was averaged over 10000 trials and a slight, constant oversampling factor of 5 was used ($B = 5\rho$). Figure 5.10(a) shows that the

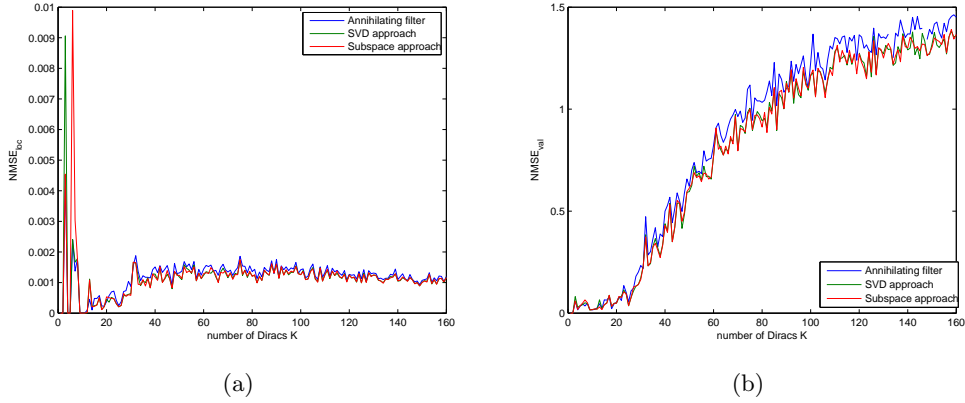


Figure 5.9: Stability of reconstruction as K increases for all presented methods (a) location error: All reconstruction methods are stable as K increases. Furthermore, the SVD and subspace approach provide slightly better results (b) amplitude error: The problem recovering amplitudes described in the annihilating filter is not solved, though the error improves slightly (code)

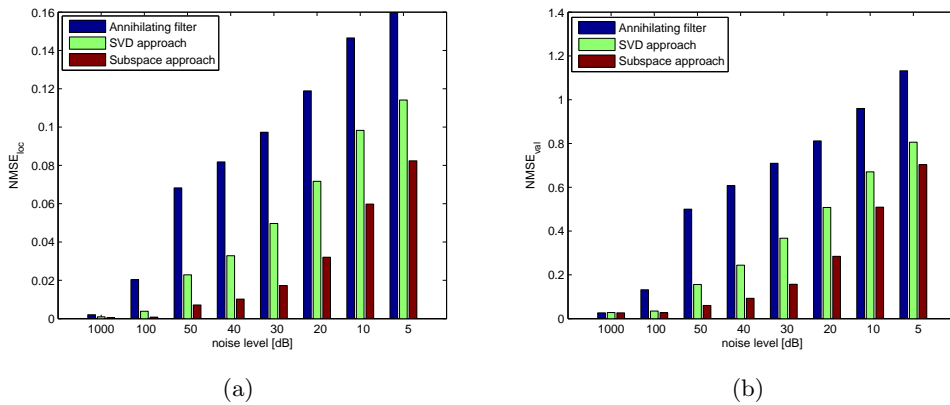


Figure 5.10: Noise robustness of all discussed recovery methods. All experiments used a bandwidth of $B = 5\rho$. (a) location error: The annihilating filter provides the worst results, the subspace approach the best results. With no oversampling, all three methods perform equal (b) The same tendency is true for the amplitude error (code)

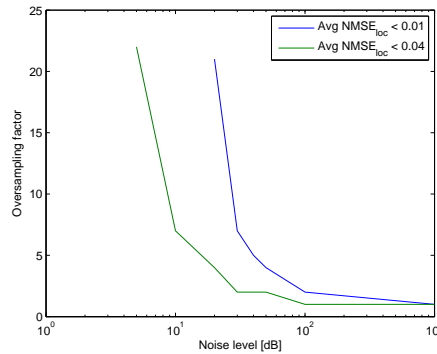


Figure 5.11: Empirical evaluation for required oversampling for a specific level of noise. While oversampling is a good option for small levels of noise (down to 20 dB), the required oversampling factor drastically increases for high levels of noise (`code`)

location error with the SVD and the subspace approach gracefully decreases. As shown in figure 5.10(b), the same tendency is true for the amplitude reconstruction. A law which describes the oversampling factor in relation to noise level is missing until now. A simple empirical approach (using the subspace approach) is shown in figure 5.11. The graph shows required oversampling factors for a specific average reconstruction error (location NMSE). This works for noise levels down to 20 dB, afterwards the required oversampling factor drastically increases. In general, the experiments show that the SVD and the subspace approach indeed improve the reconstruction performance in the noisy case.

Dirac estimation The number of Diracs K also can be estimated using the subspace approach: The rank of the matrix \mathbf{W} (5.35) provides a good estimate of the Diracs. In contrast to the method presented with the SVD approach, this method works reliable. This is shown in figure 5.12.

5.5 Closely spaced Diracs

Another question is how the reconstruction performs with closely spaced Diracs. For this reason, an experiment consisting of two Diracs at $t_1 = 0.5$ and at $t_2 = t_1 + \Delta$ was conducted. The Dirac spacing Δ was decreased logarithmically. Figure 5.13 shows the results of the experiment using the annihilating filter (figure 5.13(a)) and the subspace approach (figure 5.13(b)). In both cases, at critical sampling (using a bandwidth of $B = \rho$), the location error (NMSE_{loc}) increases gracefully with decreasing Δ . Increasing the bandwidth B (and therefore also the number of samples M) does not improve the performance in the annihilating filter case. A similar result is true for the SVD approach.

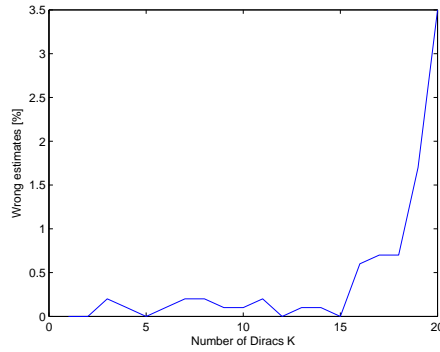


Figure 5.12: Estimating the number of Diracs K using the subspace approach: In contrast to the SVD method (cf. figure 5.8), this method works reliable (code)

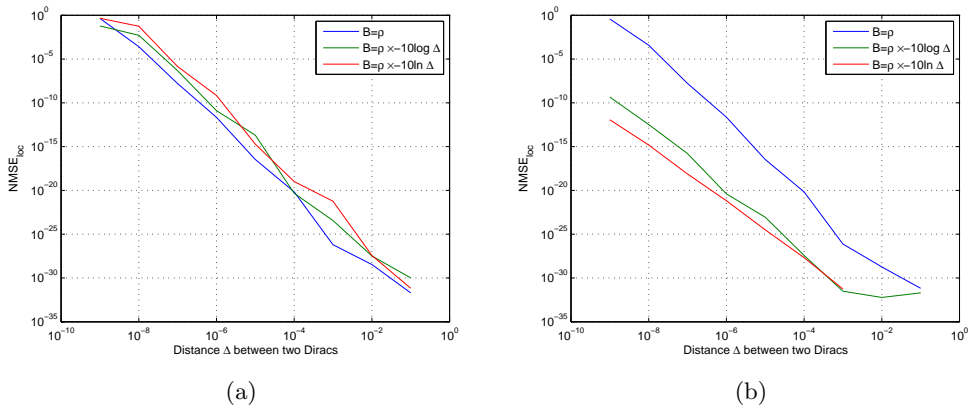


Figure 5.13: Location error in case of closely spaced Diracs. The experiment used 2 Diracs spaced Δ apart as input signal. Reconstruction was done using (a) the annihilating filter and (b) the subspace approach. At critical sampling, the location error directly increases by the spacing Δ in both cases. Oversampling helps in case of the subspace approach (code)

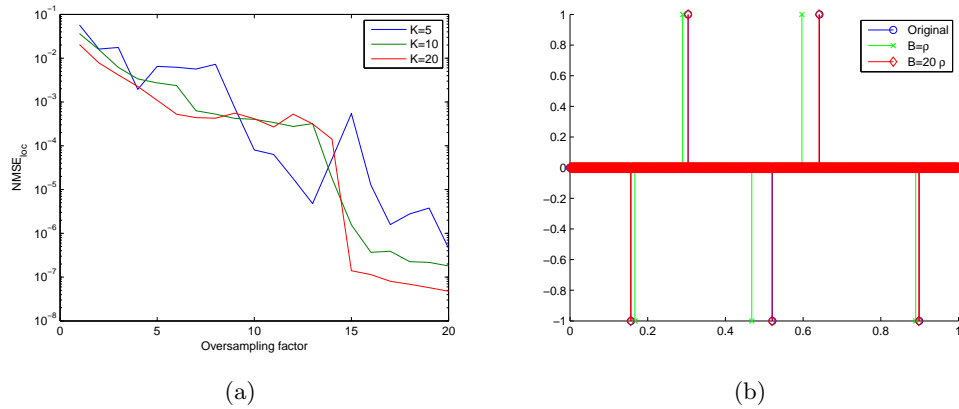


Figure 5.14: Reconstruction performance when using a Butterworth filter instead of an ideal low-pass filter (a) While the location error is huge without oversampling, additional oversampling achieves performance similar as in the idealized case (code) (b) A quantitative example confirms the results of (a) (code)

However, using the subspace approach, the location error can be reduced significantly by using a larger bandwidth. Therefore the subspace approach has better numerical stability than polynomial rooting. This fact was also described in [50]. In general it is important to avoid closely spaced Diracs because the location error is related to the Dirac spacing Δ directly.

5.6 Butterworth filter

The FRI framework presented until now has limited practical usage because it requires an ideal low-pass filter which in turn implies a sinc kernel of infinite extend. An important question is if the Fourier coefficients can be acquired by non-ideal low-pass filters such as the Butterworth filter. Figure 5.14 shows an experiment using a Butterworth low-pass filter of order 5. The Dirac stream has been simulated using a high resolution digital signal and filtered over 10 periods. The number of Diracs was chosen as $K = \{5, 10, 20\}$ and the filter bandwidth increased to a factor up to 20 from the minimum requirement. Figure 5.14(a) shows the location error for different number of Diracs K in relation to the oversampling factor. While the reconstruction error is large without oversampling, the error undoubtedly decreases as the bandwidth increases. In order to give the results a qualitative meaning, figure 5.14(b) shows a Dirac stream consisting of $K = 5$ Diracs with no oversampling and the maximum oversampling factor used in figure 5.14(a). This example suggests that a Butterworth filter can approximate the results of the sinc filter closely, if an appropriate oversampling factor is used.

On the other hand, the results are not perfectly consistent. While the tendency seems to be correct, the lines are not smooth although they have been averaged over 500 trials. Furthermore, there is no clear separation between different values of K . Another issue is amplitude reconstruction. Using the Butterworth filter, the discussed problems about amplitude reconstruction persist. Therefore, figure 5.14(b) uses only the time location results; the amplitudes have been reset to ± 1 in all cases.

5.7 Gauss kernels

While using Butterworth filters move one step closer towards a practical implementation of the FRI framework, the system still suffers from a fundamental limitation: Until now, the input signals had to be periodic. However, in practice mostly aperiodic and finite signals occur. For communication systems like UWB, aperiodic signals are essential since it is important to carry as much information as possible. By contrast, periodic signals carry only the information which is confined in one signal period T .

The authors generalize the framework presented so far for aperiodic and finite signals [31]. However, their approach is problematic for two reasons: First, the method is not based on estimating Fourier coefficients. Instead, finite differences are used in order to *annihilate* a system of polynomials. The derivation is based on the strict equivalence $\text{sinc}(x) = \sin(x)/x$ and therefore the kernel has infinite extend again. Second, the usage of the finite differences lead numerical problems as discussed in [51]. Therefore this approach is more of theoretical interest.

A better idea is to use kernels with finite support. The Gauss kernel $h(t) = e^{-t^2/2\sigma^2}$ for example has an exponential rate of decay and therefore provides a quasi-local reconstruction. In the same paper [31] it was shown how to use the Gauss kernel for reconstructing aperiodic streams of Diracs. The samples using the Gauss filter are given by:

$$y[m] = \langle h(t - mT_s), x(t) \rangle = \left\langle e^{-(t-mT_s)^2/2\sigma^2}, x(t) \right\rangle \quad (5.37)$$

From (5.1), define a finite stream of Diracs as follows:

$$x(t) = \sum_{k=1}^K c_k \delta(t - t_k) \quad (5.38)$$

Inserting (5.38) into (5.37) becomes:

$$y[m] = \sum_{k=1}^K c_k e^{-(t-mT_s)^2/2\sigma^2}$$

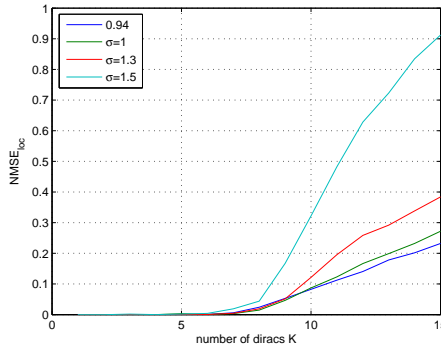


Figure 5.15: Stability of reconstruction using the Gauss kernel. Because of the exponential nature of the Gauss kernel the reconstruction is very unstable (code)

$$\begin{aligned}
 &= \sum_{k=1}^K \left(c_k e^{-t^2/2\sigma^2} \right) \cdot e^{mt_k/(\sigma^2 f_s)} \cdot e^{-m^2/(2\sigma^2 f_s^2)} \\
 \Leftrightarrow \underbrace{y[m] e^{m^2/(2\sigma^2 f_s^2)}}_{\mathbf{y}'} &= \sum_{k=1}^K \underbrace{c_k e^{-t^2/2\sigma^2}}_{c'_k} \cdot \underbrace{e^{mt_k/(\sigma^2 f_s)}}_{z'^m_k}
 \end{aligned} \tag{5.39}$$

This signal is a sum of exponentials and has the same form as (5.10). Therefore the discussed recovery strategies can be used to reconstruct the signal.

Figure 5.15 shows the stability for the Gauss kernel. While the reconstruction works for small values of K ($K < 5$), the location reconstruction error increases very fast. The exponential function leads to a very high dynamic range between the samples and therefore the matrix \mathbf{W} . For $K = 10$ for example, the dynamic range is already 10^{60} . The consequence is a very unstable reconstruction. The variance σ^2 has to be chosen very carefully [31] but no concrete details have been published. Choosing a small σ^2 provides a better recovery performance but quickly leads to infinite values. For example, when $\sigma^2 < 1$ the vector \mathbf{y}' contains infinite values. Choosing a large σ^2 provides a stable reconstruction process because the kernel support becomes larger but the reconstruction error increases even faster.

5.8 Spline kernels

5.8.1 Polynomial reproducing kernels

In [30], the FRI framework was generalized to a wide range of kernels, namely kernels which satisfy the Strang-Fix conditions [56]. The important fact about these kernels is that they are able to reproduce polynomials. That means, a linear combination of

shifted kernels is equal to a polynomial. An important class of functions having this property are B-splines [57]. A B-spline $\beta_N(t)$ of order N can reproduce polynomials up to degree N and has a support of $S = N + 1$ which is the smallest possible for this order of approximation [30]:

$$\sum_{m \in \mathbb{Z}} c_{m,n} \beta_N(t - m) = t^n, \quad n = 0, 1, \dots, N \quad (5.40)$$

Figure A.1 in appendix A shows how this formula can reproduce polynomials. Figure 5.16(a) shows a family of B-splines for $N = 0, \dots, 4$. A B-spline of order N is obtained as a repeated convolution with the box spline $\beta_0(t)$ [58]:

$$\beta_N(t) = \underbrace{(\beta_0 * \beta_0 * \dots * \beta_0)}_{N+1}(t) \quad (5.41)$$

Because of the compact support, a B-spline kernel is able to reconstruct a finite stream of Diracs as defined by (5.38) and even an infinite stream of Diracs if there are no more than K Diracs in an interval of size $2KST_s$. It is also possible to reduce piecewise polynomial signals to a stream of Diracs similar as for sinc kernels. The samples of an FRI system using a B-spline $h(t) = \beta_N(t)$ are given by:

$$y[m] = \langle \beta_N(t - mT_s), x(t) \rangle \quad (5.42)$$

Figure 5.16 shows an example of sampling $K = 4$ Diracs with the kernel β_5 . The reconstruction starts by multiplying each sample with the constants $c_{m,n}$ from (5.40) and summing them up. This generates the innovation signal $\tau[n]$. For simplicity, the derivation uses $fs = 1$:

$$\begin{aligned} \tau[n] &= \sum_m c_{m,n} \cdot y[m] \\ &= \sum_m c_{m,n} \langle x(t), \beta_N(t - m) \rangle && \text{(from (5.42))} \\ &= \left\langle x(t), \underbrace{\sum_m c_{m,n} \beta_N(t - m)}_{t^n} \right\rangle && \text{(from (5.40))} \\ &= \left\langle \sum_{k=1}^K c_k \delta(t - t_k), t^n \right\rangle && \text{(from (5.38))} \end{aligned}$$

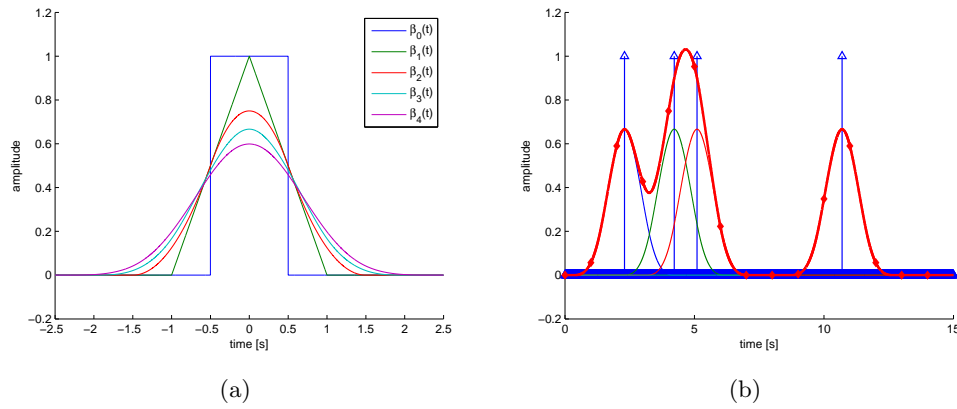


Figure 5.16: Using B-splines as kernel (a) Family of B-splines for $N = 0, \dots, 4$. The maximum support is $S = N + 1 = 5$ (code) (b) filtered signal $y(t)$ with samples $y[m]$ at rate $f_s = 1$ Hz (code)

$$\begin{aligned}
 &= \int_{-\infty}^{\infty} \sum_{k=1}^K c_k \delta(t - t_k) t^n dt \\
 &= \sum_{k=1}^K c_k \cdot \underbrace{t_k^n}_{z_k^n}
 \end{aligned} \tag{5.43}$$

The signal $\tau[n]$ consists solely of known components. Equation (5.43) is nothing more than the n -th order moment of $x(t)$. Since it has the same form as (5.10) again, the discussed recovery methods can be used to recover the signal $x(t)$. In order for the system to work, the order of the B-spline has to satisfy $N \geq 2K - 1$ where K is the number of Diracs. An open question is how to obtain the coefficients $c_{m,n}$. A derivation is presented in appendix A.

Evaluation

Oversampling In contrast to the sinc filter case, oversampling is more difficult to realize. One straight forward approach is to choose a maximum number of allowed Diracs $L > K$ which in turn generates a high order kernel and therefore a longer $\tau[n]$. Figure 5.17 shows the result of an experiment recovering $K = 2$ Diracs with L varying from K to 10. The performance of the annihilating filter does not increase since it always uses only $2K$ values of $\tau[n]$. By contrast, the SVD and subspace approach use $2L$ values. However, the performance becomes worse quickly because of the huge numbers involved. A different method to incorporate more samples is suggested in [30]: Using polyphase

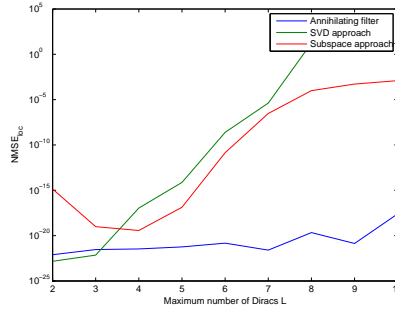


Figure 5.17: The effect of oversampling using B-splines of higher order than required. In consequence of the fast growth of the signal $\tau[n]$, the reconstruction becomes even worse for the two recovery strategies making use of the additional degrees of freedom (SVD and subspace approach) (code)

components, multiple equivalent expressions for $\tau[n]$ can be derived which are averaged afterwards. However, this method only helps in the noisy case, as the noise is averaged over all samples. Furthermore, although this estimator was shown to be asymptotically unbiased, an example in [30] suggests that the decay is not very fast.

Stability Figure 5.18(a) shows the location NMSE as the number of Diracs K increases. The experiment was conducted using K random Diracs between $t = [0, 20)$ and for each number of K , it was repeated 100 times. For the signals as well as for the kernel, a grid of 1000 points per second was used. The reconstruction works up to machine precision for $K < 3$ but afterwards the error quickly increases. The stability does not depend on the reconstruction method. Reconstruction of the amplitudes is even more problematic than in the sinc case. It works for $K = \{1, 2\}$ but afterwards the reconstructed amplitudes move towards infinity. This is because the matrix \mathbf{V}_1 in (5.27) tends to become singular very quickly. In order to avoid problems with the sampling kernel, the innovation signal $\tau[n]$ has been calculated directly from (5.43). The results are shown in figure 5.18(b). Although the results are a little bit better than in figure 5.18(a), they clearly show that using B-spline kernels is unstable by concept.

5.8.2 Exponential kernels

Even if polynomial splines would be stable there is one important question: Which real devices form a filter with a B-spline as impulse response? The answer are *exponential splines*. Recently it was shown that exponential splines (E-splines) can be used as basis functions in signal processing [59, 60]. Furthermore, a FRI system using E-splines can be linked to rational transfer functions using a digital filter [30]. Therefore any electrical circuit having a rational transfer function can be used as sampling device. Figure 5.19

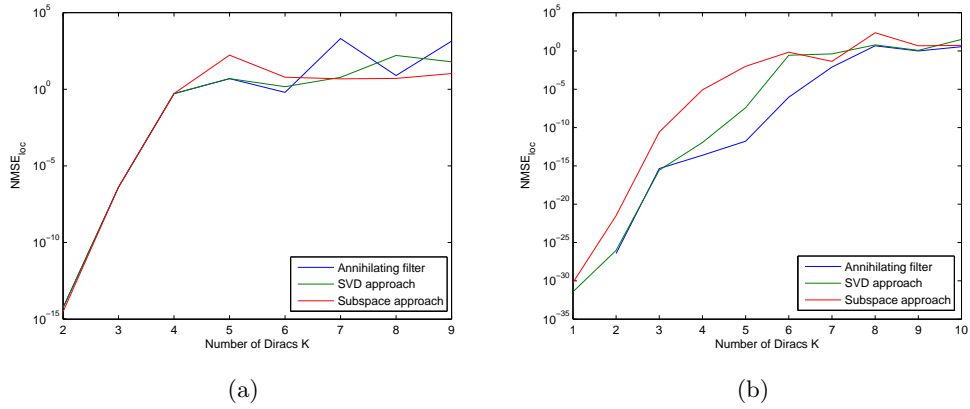


Figure 5.18: (a) Stability of the FRI framework using B-spline kernels as K increases. The system only works reliable for a very small number of K . Reconstructing the amplitudes is even more problematic (b) (code) In order to avoid effects with the simulation of the filter operation, the innovation signal was directly derived from (5.43). Even in this case the reconstruction is very unstable (code)

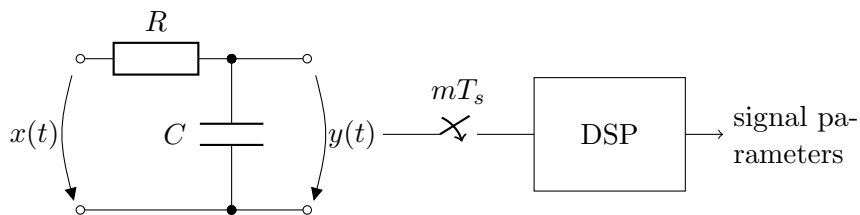


Figure 5.19: Possible usage of exponential splines: A simple RC circuit can sample a piecewise constant signal $x(t)$

shows an example of such a setup: An ordinary RC circuit is able to sample a piecewise constant signal $x(t)$. In Fourier domain, an exponential spline of order one is defined as:

$$B_\alpha(\omega) = \frac{1 - e^{\alpha - j\omega}}{j\omega - \alpha} \quad (5.44)$$

The exponential spline is dependent on the parameter α . The polynomial B-spline is a special case of the exponential spline with $\alpha = 1$. E-splines of order $N > 1$ may have different parameters for each “order”. Therefore E-splines of order $N > 1$ are described using a vector $\mathbf{a} = \{\alpha_0, \alpha_1, \dots, \alpha_N\}$:

$$B_{\mathbf{a}}(\omega) = \prod_{n=1}^N \frac{1 - e^{\alpha_n - j\omega}}{j\omega - \alpha_n} \quad (5.45)$$

The principle is similar as for the polynomial case. The kernel has to reproduce exponentials of the form $e^{(\alpha_0+n\lambda)t} = e^{\alpha_n t}$ for $n = 0, 1 \dots N$ and the kernel must be of order $N \geq 2K - 1$:

$$\sum_{m \in \mathbb{Z}} c_{m,n} \beta_{\mathbf{a}}(t - m) = e^{(\alpha_0+n\lambda)t} \quad (5.46)$$

As for the polynomial case, the coefficients $c_{m,n}$ have to be found for a specific kernel. A derivation for calculating these coefficients is given in appendix B. Using the same derivation as for the polynomial case, an innovation signal $s[n]$ is found:

$$s[n] = \sum_{m \in \mathbb{Z}} c_{m,n} y[m] = \dots = \sum_{k=1}^K \underbrace{c_k e^{\alpha_0 t_k}}_{c'_k} \cdot \underbrace{e^{(\lambda t_k)^n}}_{z_k^n} \quad (5.47)$$

Evaluation

In general the discussion for the B-splines applies for the E-splines as well. Unfortunately the FRI framework using E-splines is even more instable than with B-splines. Due to the exponential nature of (5.47), the numbers become huge very quickly. Even in the absence of any sampling, the reconstruction works only for $K \leq 3$. The parameters λ and a_0 suggest a tuning parameter for the stability. However, the error does not decrease perceptibly with any specific set of $\{\lambda, a_0\}$ and there is no noticeable pattern with which the optimal parameters could be determined. In general $a_0 = 0$ seems to be a good choice. Furthermore, the framework is very sensitive to non-ideal kernels which is a big problem when implementing real devices using rational transfer functions.

5.9 Conclusion

Sampling at the rate of innovation is an interesting concept. First, because it deals with *real* analog signals and second because it supports to sample at the minimum rate (the information content). In [48], the FRI framework was applied to a UWB receiver. However, the derivation uses the periodic stream of Diracs and an ideal low-pass filter which is not a realistic assumption. Apart from the somehow unpractical assumption, FRI using a periodic stream of Diracs works quite well. In the periodic case, the ideal low-pass filter can be approximated by a Butterworth filter at the cost of oversampling.

However, sampling non-periodic signals, either finite or infinite, is still an unresolved issue. One way to overcome this problem is to use sampling kernels with compact support. Unfortunately the Gaussian kernel as well as the moments-estimating methods are too unstable for practical usage. This fact is especially sad for the E-splines since they would enable building the sampling kernels in practice easily.

Another promising approach is presented in [61]. This approach tries to combine the reconstruction stability using Fourier coefficients on the one hand and local reconstruction using compact filters termed *Sum-of-Sinc filters* on the other hand. A multichannel extension is presented in [62]. This approach directly obtains the Fourier coefficients using analog integration. However, this requires at least $2K$ channels which limits the practical usage.

Chapter 6

The modulated wideband converter (MWC)

The MWC is the third sub-Nyquist sampling device covered by this thesis. It first was presented in [63] as an extension to [64]. Figure 6.1 shows the principal setup. First the input signal is split into M channels. Then each channel is modulated by a periodic waveform $p_i(t)$ and afterwards filtered by an analog filter $h(t)$ which is a low-pass filter with cutoff $f_s/2$. Afterwards each channel is sampled at rate f_s , resulting in M sequences $y[n]$ denoted by the vector $\mathbf{y}[n]$.

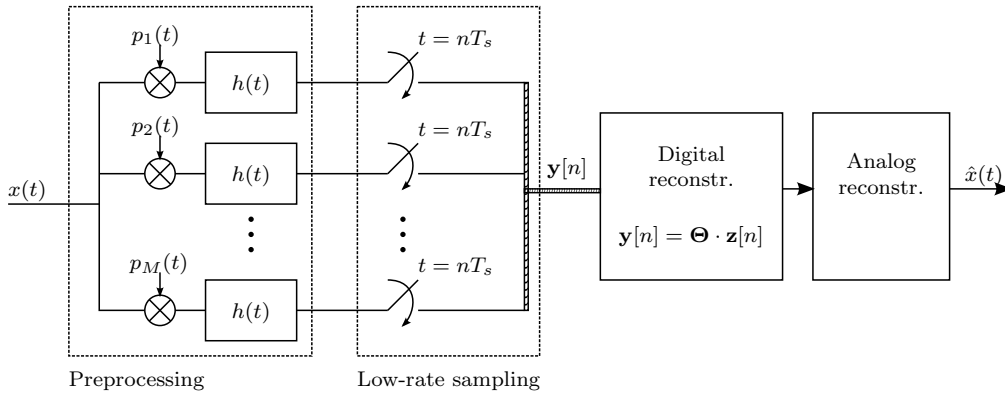


Figure 6.1: The MWC consists of M channels. Each channel first is multiplied by a periodic mixing function $p(t)$, filtered by a low-pass filter with cutoff at $f_s/2$ and afterwards sampled at a low rate f_s . The output are M sequences $y[n]$

The preprocessing and sampling stage can be regarded as a generalization of the random demodulator (RD) (see chapter 4) to M branches. In contrast to the RD, the mixing functions are periodic with period T_p . In order to meet specific compressive sampling (CS) requirements, each period additionally consists of a piecewise constant random pattern. The principle of the RD is to spread the information content of each frequency tone across the whole spectrum, whereas the MWC uses the multiplication to alias specific frequency ranges to the baseband simultaneously. The multichannel architecture suggests another principle of the MWC: The CS matrix is implemented in hardware. The M channels can be regarded as the M CS measurements.

6.1 Working principle

6.1.1 Compressive sampling for analog signals

The major problem of the RD is the application of a discrete-time signal model to analog signals. The vector \mathbf{s} in (4.18) describes the spectrum of the analog signal as a finite, discrete vector with cardinality f_N (the Nyquist rate). Application of CS works by assumption that \mathbf{s} is sparse:

$$\mathbf{y} = \Theta \cdot \mathbf{s} \quad \Rightarrow \quad \mathbf{s} = \min_{\mathbf{s}} \|\mathbf{s}\|_1 \quad \text{s.t.} \quad \mathbf{y} = \Theta \cdot \mathbf{s} \quad (6.1)$$

By contrast, analog signals have a continuous spectrum. Therefore it would be a better idea to define the sparsity constraint across a continuous range of the spectrum $X(f)$ of the input signal $x(t)$. While sparsity is defined in the discrete case (the number of non-zeros of a finite vector) easily, it is difficult in the continuous case. Putting a continuous Fourier spectrum into a vector would result in an infinite vector (with infinite cardinality). One way to overcome this problem is to assume a continuous block sparsity. In this case there are continuous non-zero blocks on the continuous spectrum. This is also how analog multiband signals look like: Denote by K the number of non-zero blocks on the continuous spectrum (number of bands), then there are $K/2$ transmissions (due to the complex conjugate). Figure 6.2(a) shows the spectrum of a wideband multiband signal with a single transmission and therefore $K = 2$. The Nyquist rate is $f_N = 10$ GHz. This spectrum can be represented in vector form by dividing it into N pieces. In the remainder, N is assumed to be an odd number with $N = 2N_0 + 1$ for simplicity. Using this definition, the spectrum slices can be stacked together in order to form a finite vector where the elements are functions of the continuous variable f :

$$\mathbf{z}(f) = \begin{bmatrix} z_{-N_0}(f) \\ z_{-N_0+1}(f) \\ \vdots \\ z_0(f) \\ \vdots \\ z_{N_0}(f) \end{bmatrix} \quad \text{with} \quad z_i(f) = X(f), \quad \frac{f_N}{N}i \leq f < \frac{f_N}{N}(i+1) \quad (6.2)$$

This is illustrated in figure 6.2. Call the bandwidth of an element in $\mathbf{z}(f)$ f_s , in this case $f_s = f_N/N$. For the remainder, define this frequency range as:

$$\mathcal{F}_s := \left[-\frac{f_s}{2}, \frac{f_s}{2} \right] \quad (6.3)$$

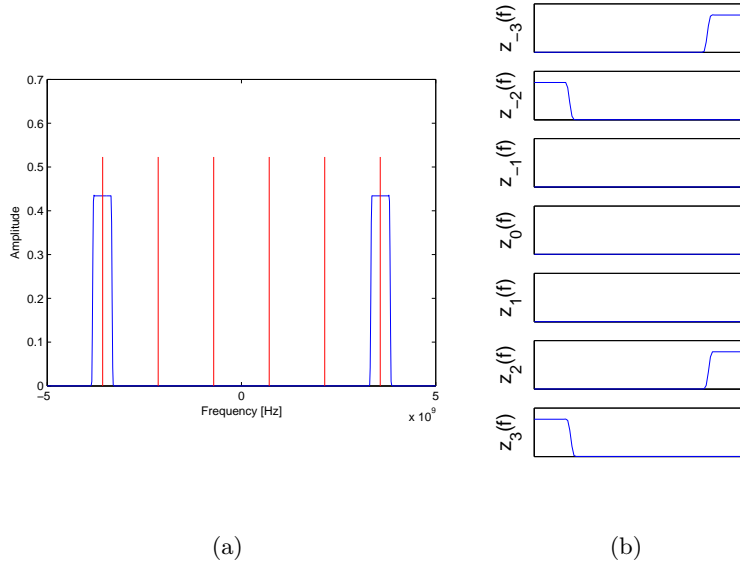


Figure 6.2: (a) Spectrum of a multiband signal. In this example, the signal contains a single transmission (b) The spectrum can be represented as a vector $\mathbf{z}(f)$ by dividing the frequency band into N slices and stacking them together. Each vector element $z_i(f)$ is dependent on the continuous variable f and $\mathbf{z}(f)$ is K -sparse for every f

Now assume an $M \times N$ sensing matrix Φ which takes the vector $\mathbf{z}(f)$ and assigns a compressed output $\mathbf{y}(f)$ (M “measurements”):

$$\mathbf{y}(f) = \Phi \cdot \mathbf{z}(f) \quad (6.4)$$

Usually such a setup is referred to as an infinite measurement vector (IMV) [65]. Naïve application of CS knowledge in order to solve (6.4) results in the following optimization problem:

$$\min_{\mathbf{z}(f)} \|\mathbf{z}(f)\|_1 \quad \text{s.t.} \quad \mathbf{y}(f) = \Phi \cdot \mathbf{z}(f) \quad (6.5)$$

While (6.1) can be solved easily using CS techniques, the problem in (6.4) depends on the continuous variable f and would require to solve infinitely many CS problems jointly. Therefore (6.4) can not be solved directly.

The authors proposed a different scheme for solving (6.5) consisting of a two-step flow which they term continuous-to-finite converter (CTF) [65, 64]. First, denote with $\text{supp}(\mathbf{a})$

of the vector \mathbf{a} the support or the indexes which are non-zero. This is equivalent with the sparsity pattern. The vector $\mathbf{z}(f)$ can be regarded as a collection of infinitely many vectors (one for each value of $f \in \mathcal{F}_s$), which jointly have the same sparsity pattern. Therefore the support of $\mathbf{z}(f)$ can be defined as the union over all vectors for a *all* possible values of f [64]:

$$\mathcal{S} := \text{supp}(\mathbf{z}(\mathcal{F}_s)) = \bigcup_{f \in \mathcal{F}_s} \text{supp}(\mathbf{z}(f)) \quad (6.6)$$

For example, in figure 6.2, the vector $\mathbf{z}(f)$ is 2-sparse for every value of $f \in \mathcal{F}_s$. By contrast, the union over the collection of all values $f \in \mathcal{F}_s$ is 4-sparse because the two bands contribute to four spectrum slices.

Suppose that the support \mathcal{S} is known, then (6.5) can be solved by noting that (6.4) can be written as [64]:

$$\mathbf{y}(f) = \Phi_{\mathcal{S}} \cdot \mathbf{z}_{\mathcal{S}}(f) \quad (6.7)$$

where the vector $\mathbf{z}_{\mathcal{S}}(f)$ contains only the non-zero elements of $\mathbf{z}(f)$ and $\Phi_{\mathcal{S}}$ contains only those columns of Φ which contribute to $\mathbf{y}(f)$. The equivalence is true because the elements consisting of zeros in $\mathbf{z}(f)$ do not contribute to the result in $\mathbf{y}(f)$. Given this equivalence, (6.7) can be solved using the pseudo-inverse easily:

$$\mathbf{z}_{\mathcal{S}}(f) = \Phi_{\mathcal{S}}^{\#} \cdot \mathbf{y}(f) \quad (6.8)$$

Once $\mathbf{z}_{\mathcal{S}}(f)$ is found, $\mathbf{z}(f)$ can be obtained by filling all indexes which are not contained in \mathcal{S} with zeros. The remaining problem is how to obtain the support \mathcal{S} . The solution is to deduce a finite set of vectors which have the same column span as $\mathbf{y}(f)$ [65]. A matrix \mathbf{V} with these finite set of vectors as columns is called a frame for $\mathbf{y}(f)$. Since \mathbf{V} is a frame for $\mathbf{y}(f)$, there exists a finite collection of sparse vectors (represented as columns of a matrix \mathbf{U}), which have the same joint-sparsity as $\mathbf{z}(f)$:

$$\mathbf{V} = \Phi \cdot \mathbf{U} \quad (6.9)$$

The matrix \mathbf{V} can be calculated using a matrix \mathbf{Q} :

$$\mathbf{Q} = \int_{f \in \mathcal{F}_s} \mathbf{y}(f) \cdot \mathbf{y}^H(f) \, df \quad (6.10)$$

From the matrix \mathbf{Q} , the frame \mathbf{V} can be found using an eigenvalue decomposition as:

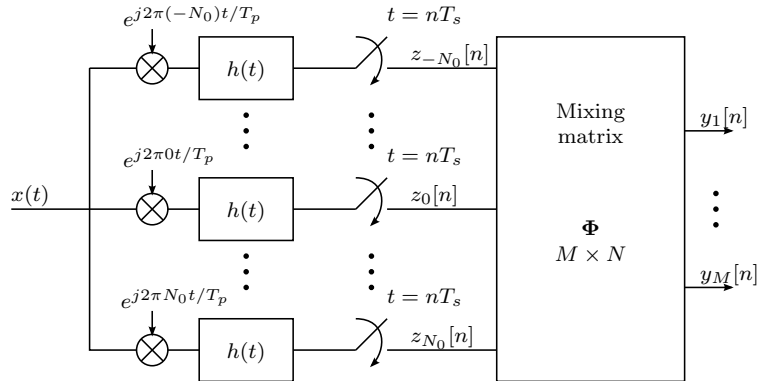


Figure 6.3: Equivalent diagram of the MWC: The vector $\mathbf{z}(f)$ is generated by aliasing the $N = 2N_0 + 1$ spectrum slices to the baseband by modulation. After sampling, the discrete representation $\mathbf{z}[n]$ is obtained. The samples $\mathbf{y}[n]$ correspond to a multiplication with a mixing matrix Φ

$$\mathbf{V} \cdot \mathbf{V}^H = \mathbf{Q} \quad (6.11)$$

Since \mathbf{V} is a (finite) frame for $\mathbf{y}(f)$, the finite set of vectors in \mathbf{U} provides the same joint-sparsity pattern as $\mathbf{z}(f)$. Thus, seeking for the sparsest collection of vectors \mathbf{U} in (6.9) provides their joint-sparsity which is the same as \mathcal{S} for $\mathbf{z}(f)$.

Equation (6.9) is termed multiple measurement vector (MMV) [66, 67] and is a generalization of a single measurement vector (SMV) as in (6.1) to more than a single vector. An MMV system is a special case of a block-sparse SMV system [68]. Several methods exist for solving MMV systems including methods based on convex relaxation [69] and greedy algorithms [70]. Another method is the *Reduce and Boost* algorithm presented in [65].

6.1.2 The equivalent diagram of the MWC

Using a continuous CS model like (6.4) is an interesting concept compared to the discrete approach like (6.1). Until now, there are two open questions: First, which sampling device can be used to obtain a model like (6.4). Second, a sampling device will produce a finite, discrete sequence. The CS matrix Θ is a sampling operator which maps a continuous function to a finite number of discrete sequences:

$$\mathbf{y}[n] = \Theta\{\mathbf{x}(t)\} \quad (6.12)$$

The answer to both questions is the MWC. The M CS measurement sequences are implemented by the M sampling channels and the CS matrix is shifted to the analog

domain. In order to understand the action of the MWC, it is instructive to look at the equivalent system [35] and use time domain analysis to show the equivalence to figure 6.1. The equivalent system is depicted in figure 6.3. Recall from section 6.1.1 that the spectrum is partitioned into N equally spaced spectrum slices. The equivalent system has N channels. To obtain the vector $\mathbf{z}(f)$ in hardware, each channel is first modulated by $\exp(-j2\pi lt/T_p)$ with $l \in \{-N_0, \dots, N_0\}$. This modulation shifts each of the N spectrum slices to the baseband of the spectrum between $\left[-\frac{1}{2T_p}, \frac{1}{2T_p}\right]$ Hz. Call this frequency range \mathcal{F}_p . With $f_p = 1/T_p$, the vector $\mathbf{z}(f)$ therefore is valid for $f \in \mathcal{F}_p$ and the width of the spectrum slices is f_p .

In order to leave the baseband ($f \in \mathcal{F}_p$) only, a low-pass filter cuts at the frequency $f_s/2 = f_p/2$. Since this signal is bandlimited now to $f_s/2$, it can be sampled at a low rate f_s to obtain N sequences $z_l[n]$:

$$z_l[n] = (x(t)e^{-j2\pi lt/T_p}) * h(t) \Big|_{t=n \cdot T_s} \quad (6.13)$$

Note that these sequences represent the input signal $x(t)$ in time domain for each spectrum slice. The sequences are complex in general because of the complex modulation. The resulting vector $\mathbf{z}[n]$ now fully describes the vector $\mathbf{z}(f)$ and the other way round. They also share the same sparsity pattern. If the input signal $x(t)$ is a sparse multiband signal (i.e., there are only a few active bands), the spectrum $X(f)$ is sparse. Therefore most of the sequences $z_l[n]$ are equal to zero and therefore \mathbf{z} is sparse. Finally, the sample sequences $y_i[n]$ are given by applying the mixing matrix Φ :

$$y_i[n] = \sum_{l=-N_0}^{N_0} \phi_{il} \cdot z_l[n] = \sum_{l=-N_0}^{N_0} \phi_{il} \cdot \left((x(t)e^{-j2\pi lt/T_p}) * h(t) \Big|_{t=n \cdot T_s} \right) \quad (6.14)$$

which can be written in vector-/matrix form using the $M \times N$ matrix Φ :

$$\mathbf{y}[n] = \Phi \cdot \mathbf{z}[n] \quad (6.15)$$

Back to the principal setup of the MWC, the mixing functions $p_i(t)$ in figure 6.1 are periodic with period T_p [63]:

$$p_i(t) = \sum_{l=-\infty}^{\infty} P_i[l] e^{j\frac{2\pi}{T_p} lt} \quad (6.16)$$

For this reason, they can be represented using their Fourier series:

$$P_i[l] = \frac{1}{T_p} \int_0^{T_p} p_i(t) e^{-j \frac{2\pi}{T_p} lt} dt \quad (6.17)$$

According to figure 6.1, the output of a single channel is given by first multiplying $x(t)$ with the mixing function $p_i(t)$ and afterwards filtering with the filter $h(t)$. Finally, the signal is sampled at rate f_s . Therefore the samples for one channel are given by:

$$y_i[n] = (x(t) \cdot p_i(t)) * h(t) \Big|_{t=n \cdot T_s} \quad (6.18)$$

Inserting (6.16) into (6.18) yields:

$$y_i[n] = \sum_{l=-\infty}^{\infty} P_i[l] \left((x(t) \cdot e^{j \frac{2\pi}{T_p} lt}) * h(t) \Big|_{t=n \cdot T_s} \right) \quad (6.19)$$

This result is essentially the same as (6.14). The Fourier coefficients $P_i[l]$ must match the entries ϕ_{il} of the mixing matrix and $p_i(t)$ must be represented with about N Fourier coefficients. This means that the representation of the mixing functions through the Fourier series offer additional degrees of freedom in order for the equivalence to work. The mixing functions $p_i(t)$ therefore implement the CS sensing matrix Φ in hardware. Since $\mathbf{z}[n]$ has the same sparsity pattern as $\mathbf{z}(f)$, the sparsity inducing matrix Ψ is the identity matrix: $\Psi = \mathbf{I}$. Therefore, for the MWC, the CS matrix Θ is the same as the sensing matrix Φ .

The idea of the sampling system is to alias certain band regions to the baseband in order to obtain a vector $\mathbf{z}(f)$ as in figure 6.2(b). Therefore, the system is applicable to multiband signals only.

6.1.3 Parameters of the sampling system

For the basic case, the period of the mixing functions T_p equals to the sampling period T_s . In order for $\mathbf{z}(f)$ to be K -sparse for every f , the maximum bandwidth of a band must not exceed f_p . Since the $p_i(t)$ sequences correspond to the CS matrix, the sequences must be chosen such that their Fourier coefficients provide specific CS properties: The matrix Φ has the size $M \times N$. This means that the number of Fourier coefficients must be about N and the rows of Φ must be linearly independent. In particular, this is true for piecewise constant functions which alternate between ± 1 at a high rate of at least $f_p N$ [64]:

$$p_i(t) = \alpha_{ik}, \quad t \in \left[k \cdot \frac{T_p}{N}, (k+1) \cdot \frac{T_p}{N} \right], \quad \alpha_{ik} \in \{-1, 1\}, \quad 0 \leq k \leq N-1 \quad (6.20)$$

The Fourier coefficients of the $p_i(t)$ in (6.20) can be found from (6.17) using the shift theorem:

$$\begin{aligned}
 P_i[l] &= \frac{1}{T_p} \int_0^{\frac{T_p}{N}} \sum_{k=0}^{N-1} \alpha_{ik} e^{-j\frac{2\pi}{T_p}lk\frac{T_p}{N}} e^{-j\frac{2\pi}{T_p}lt} dt \\
 &= \sum_{k=0}^{N-1} \alpha_{ik} e^{-j\frac{2\pi}{N}lk} \frac{1}{T_p} \int_0^{\frac{T_p}{N}} e^{-j\frac{2\pi}{T_p}lt} dt \\
 &= \sum_{k=0}^{N-1} \alpha_{ik} \cdot e^{-j\frac{2\pi}{N}lk} \cdot \underbrace{\frac{1 - e^{-j\frac{2\pi}{N}l}}{j2\pi l}}_{d_l}
 \end{aligned} \tag{6.21}$$

For a specific channel i , $P_i[l]$ corresponds to the N -point discrete Fourier transform (DFT) of the sequence $\alpha_i[k]$ where each element is multiplied by d_l . This can be written in matrix form: $\mathbf{p}_i = \mathbf{a}_i \cdot \mathbf{F} \cdot \text{diag}(d_l)$ where \mathbf{F} is the DFT matrix and contains the entries $\exp(-j2\pi lk/N)$. For all M channels together, this can further be written as

$$\mathbf{P} = \mathbf{A}\mathbf{F}\mathbf{D} = \mathbf{\Phi} \tag{6.22}$$

where $\mathbf{D} = \text{diag}(d_l)$ is the diagonal matrix containing the last term in (6.21). The last equivalence follows from the fact that $\mathbf{\Phi}$ contains the Fourier coefficients $P_i[l]$ (cf. section 6.1.2).

The matrix \mathbf{A} contains the sign patterns α_{ik} of the functions $p_i(t)$ and dictates the CS properties of the CS matrix directly: \mathbf{A} may be a random sign matrix, drawn from a Gaussian distribution (cf. section 2.2). Thus, the sequences $p_i(t)$ can be implemented by a shift register easily.

The number of channels also is dictated by CS properties, since it corresponds to the number of measurements: It must be high enough such that the MMV system in (6.9) can be solved. Similar to an SMV system, the number of channels must obey [63]:

$$M \approx 4K \log\left(\frac{N}{2K}\right) \tag{6.23}$$

This result stems from the following observation: Solving (6.4) in polynomial time for every f would require $M = \mathcal{O}(K \log(N/K))$ measurements. Since $\mathbf{\Phi}$ requires every $2K$ columns to be linearly independent in order for \mathbf{z} to be unique K -sparse, M must be at least $M \geq 2K \log(N/K)$. However, $\mathbf{z}(f)$ is jointly $2K$ -sparse and not K -sparse. The CTF reduces $\mathbf{z}(f)$ to the finite \mathbf{U} which is also jointly $2K$ -sparse. The reduction to the MMV system only preserves the joint-sparsity, therefore \mathbf{U} might contain entries

which are jointly $2K$ -sparse. For this reason, the reduction by CTF deals with $2K$ -sparse vectors and requires the number of channels to be doubled. Since the system consists of M channels given by (6.23) and each channel samples at rate f_s , the total sampling rate is given by:

$$f_{\text{tot}} = f_s \cdot M \approx f_s 4K \log\left(\frac{N}{2K}\right) \quad (6.24)$$

This number only depends on the number of transmissions ($K/2$) but not on the Nyquist rate.

6.1.4 Reconstruction

The reconstruction consists of two parts: A digital part which involves finding the correct support \mathcal{S} and recovering $\mathbf{z}[n]$. An optional analog part produces $\hat{x}(t)$ from $\mathbf{z}[n]$.

Digital reconstruction

The heart of the digital reconstruction is the CTF: It takes the M signal sequences $\mathbf{y}[n]$ and first obtains a finite frame by using (6.10). In contrast to the continuous $\mathbf{y}(f)$, the $\mathbf{y}[n]$ are discrete sequences. Therefore the integral can be replaced by a sum [64]:

$$\mathbf{Q} = \sum_n \mathbf{y}[n] \cdot \mathbf{y}^H[n] \quad (6.25)$$

In theory, this involves an infinite number of samples. However, in practice, a set of $2K$ linearly independent vectors (and therefore samples in $\mathbf{y}[n]$) suffices [64]. The reason is that \mathbf{U} in the MMV (6.9) is jointly $2K$ -sparse and therefore $\text{rank}(\mathbf{V}) \leq 2K$. Thus, $2K$ eigenvectors in the eigenvalue decomposition (6.11) are enough to construct \mathbf{V} . The matrix \mathbf{Q} is quadratic by the number of channels and therefore typically small. For the reason discussed above, the eigenvalue decomposition only keeps $2K$ eigenvectors, therefore \mathbf{V} is small as well: $M \times 2K$.

In the next step, the MMV in (6.9) is solved in order to obtain \mathcal{S} . This is the only non-linear step in the reconstruction stage. Finally, $\mathbf{z}_S[n]$ are found from the linear system (6.8). Since $\mathbf{z}[n]$ fully describes $\mathbf{z}(f)$ and share the same sparsity, $\mathbf{z}_S[n]$ is obtained by

$$\mathbf{z}_S[n] = \Phi_S^\# \cdot \mathbf{y}[n] \quad (6.26)$$

Apart from matrix multiplications in the range $2K \times M$, this step only involves inverting a matrix with dimensions $2K \times 2K$.

The digital reconstruction is summarized in figure 6.4.

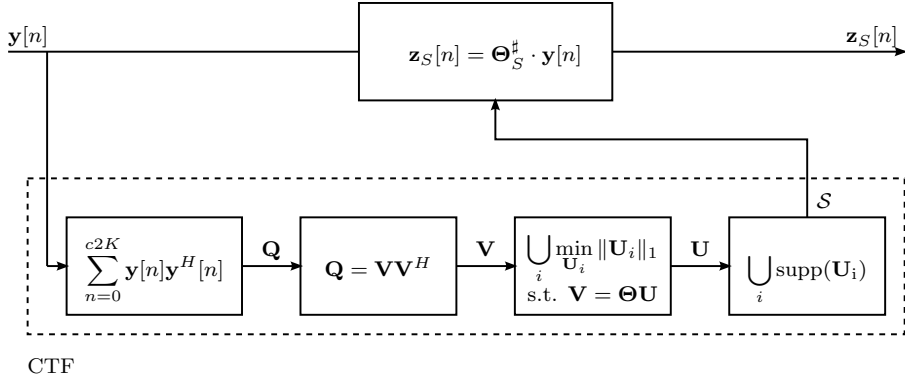


Figure 6.4: Digital reconstruction of the MWC: In general, the signal is reconstructed by solving a linear system. The non-linear CTF part only is required for determining the support of the input signal

Analog reconstruction

There are two ways for obtaining the analog $\hat{x}(t)$ [64]. The first option is to reconstruct the signal at the Nyquist rate. Each of the samples in $\mathbf{z}_S[n]$ is expanded and digitally interpolated to the Nyquist rate f_N . Afterwards, these samples are digitally modulated to their respective locations. Finally, these samples are fed into a digital-to-analog converter (DAC) at Nyquist rate. This option is reliable for small Nyquist rates only.

The second option is to shift the modulation to the analog domain: Each sequence $z_i[n]$ is fed into a DAC at rate f_p . $K/2$ modulators modulate the bands to their respective locations in hardware:

$$\hat{x}(t) = \sum_{i \in \mathcal{S}, i \geq 0} \Re\{z_i(t)\} \cos(2\pi i f_p t) + \Im\{z_i(t)\} \sin(2\pi i f_p t) \quad (6.27)$$

With this approach, the sampling system does not need to operate at the Nyquist rate. However, it requires $N/2$ modulators with non-constant frequencies at $i \cdot f_p$.

6.1.5 Reduction of channels

The MWC presented so far requires at least eight times as many channels as number of transmissions (cf. (6.23)). In this case, the periods T_p of the modulation sequences $p_i(t)$ equal to the sampling periods T_s of the low-rate analog-to-digital converter (ADC). The authors presented an approach [64] which decreases the number of channels at the cost of a higher sampling rate $f_s = q \cdot f_p$ (with odd $q = 2q' + 1$) in each channel. This approach is best understood in the frequency domain. Consider (6.14) in the frequency domain:

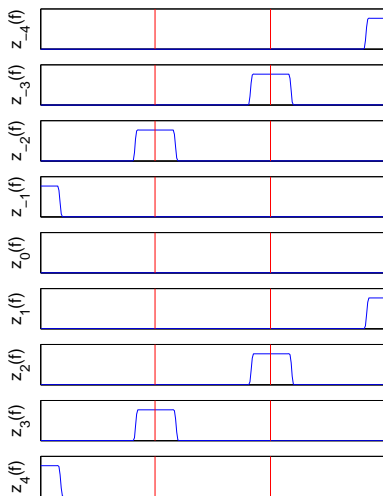


Figure 6.5: The vector $\mathbf{z}'(f)$ when $f_s = 3f_p$. All bands must contribute to $\mathbf{z}'(f)$. Therefore $N' = N + (q - 1)$ in this construction. The middle part is equal to $\mathbf{z}(f)$ in figure 6.2(b) (except $z_{\pm 4}(f)$ which are zero and not existent in figure 6.2(b))

$$y_i(f) = \sum_{l=-N_0}^{N_0} \phi_{il} \cdot z_l(f), \quad f \in \mathcal{F}_s \quad (6.28)$$

As in section 6.1.1, this relation is valid in \mathcal{F}_s . For the MWC implementation, this corresponds to the range $[-f_s/2, f_s/2]$ because each channel sequence is low-pass filtered by a filter with cutoff at $f_s/2$. According to (6.14) and figure 6.2, $z_l(f)$ corresponds to f_p -shifted spectrum slices $X(f - lf_p)$:

$$y_i(f) = \sum_{l=-N_0}^{N_0} \phi_{il} \cdot X(f - lf_p), \quad f \in \mathcal{F}_s \quad (6.29)$$

Until now, the spectrum was partitioned into N slices with bandwidth f_p . When $f_p > f_s$, the number of entries in $\mathbf{z}(f)$ becomes larger. Denote the vector with $\mathbf{z}'(f)$ in this case and the output vector with $\mathbf{y}'(f)$. Figure 6.5 shows an example of $\mathbf{z}'(f)$ with $f_s = 3f_p$ (corresponding to the spectrum in figure 6.2(a)). The frequency range contained in $\mathbf{z}'(f)$ now is three times larger than with $f_p = f_s$. Since this frequency range

is determined by the low-pass filter with cutoff at $f_s/2$ only, the middle part is equal to figure 6.2(b). However, a necessary condition for uniqueness is that each band contributes to $\mathbf{z}'(f)$. In the simple case $f_s = f_p$, this is ensured by setting $N = 2N_0 + 1 = f_N/f_p$ (see figure 6.2(b)). In the case when $f_s = qf_p$, the size of $\mathbf{z}'(f)$ becomes:

$$N' = N + (q - 1) \quad (6.30)$$

where N is the vector size for $f_s = f_p$. This stems from the fact that each band must be included in $\mathbf{z}'(f)$.

A look at figure 6.5 suggests that fewer channels might be necessary in this case because each $z'_i(f)$ contains more information for $f_s = qf_p$ than for $f_s = f_p$. Indeed, this is the case. The idea is to recover the vector $\mathbf{y}(f)$ which would have been obtained if $\mathbf{z}(f)$ with $f_s = f_p$ from figure 6.2(b) had been used. First note that the bandwidth of $\mathbf{y}'(f)$ equals the bandwidth of $\mathbf{z}'(f)$. That is, the bandwidth of $\mathbf{y}'(f)$ is qf_p (over \mathcal{F}_s) as well. The derivation starts from (6.29). Let $f = f' + kf_p$ where $-q' \leq k \leq q'$ such that $f \in \mathcal{F}_s$ and $f_s = (2q' + 1)f_p$ in order to alias all $f \in \mathcal{F}_s$ of $\mathbf{y}'(f)$ to $f \in \mathcal{F}_p$. Inserting this f into (6.29) yields:

$$\begin{aligned} y'_i(f' + kf_p) &= \sum_{l=-N_0}^{N_0} \phi_{il} X(f' + kf_p - lf_p) \\ &= \sum_{l=-N_0}^{N_0} \phi_{il} X(f' - \underbrace{(l-k)}_{l'} f_p), \quad \begin{cases} l' = l - k \\ l = l' + k \end{cases} \\ &= \sum_{l'+k=-N_0}^{N_0} \phi_{i,(l'+k)} X(f' - l' f_p) \\ &= \sum_{l'=-N_0-k}^{N_0-k} \phi_{i,(l'+k)} X(f' - l' f_p) \end{aligned} \quad (6.31)$$

$X(f' - l' f_p)$ corresponds to the spectrum slices with $f \in \mathcal{F}_p$, that is, the middle of figure 6.5. As can be seen from this figure, the middle part is always zero for the first q' and the last q' entries. Therefore (6.31) can be rewritten as:

$$y'_i(f' + kf_p) = \sum_{l'=-N_0}^{N_0} \phi_{i,(l'+k)} X(f' - l' f_p) \quad (6.32)$$

For each channel, (6.32) provides q equations for $\mathbf{z}(f)$ over \mathcal{F}_p . Thus, the whole system has Mq equations for $\mathbf{z}(f)$. The set of Mq equations can be written in Matrix notation with the $(Mq \times N)$ matrix Φ as:

$$\underbrace{\begin{bmatrix} y'_0(f - qf_p) \\ \vdots \\ y'_0(f) \\ \vdots \\ y'_0(f + qf_p) \\ \vdots \\ y'_M(f - qf_p) \\ \vdots \\ y'_M(f + qf_p) \end{bmatrix}}_{\mathbf{y}(f)} = \underbrace{\begin{bmatrix} \phi_{0,-N_0-q'} & \cdots & \phi_{0,N_0-q'} \\ \vdots & \ddots & \vdots \\ \phi_{0,-N_0} & \cdots & \phi_{0,N_0} \\ \vdots & \ddots & \vdots \\ \phi_{0,-N_0+q'} & \cdots & \phi_{0,N_0+q'} \\ \vdots & \ddots & \vdots \\ \phi_{M,-N_0-q'} & \cdots & \phi_{M,N_0-q'} \\ \vdots & \ddots & \vdots \\ \phi_{M,-N_0+q'} & \cdots & \phi_{M,N_0+q'} \end{bmatrix}}_{\Phi} \cdot \begin{bmatrix} | \\ | \\ \mathbf{z}(f) \\ | \\ | \end{bmatrix}, \quad f \in \mathcal{F}_p \quad (6.33)$$

The number of required equations is given by equation (6.23) (and was set equally to the number of channels previously because $f_s = f_p$). Therefore in the case of $f_s = qf_p$, the number of channels can be reduced to:

$$M \approx \frac{4K}{q} \log \left(\frac{N}{2K} \right) \quad (6.34)$$

In words, a q times higher sampling rate in each channel means a channel reduction by the factor of q .

There is one issue left: (6.32) aliases all relevant information to \mathcal{F}_p . However, everything outside this frequency range needs to be set to zero the same way as explained for the equivalent system. This can be done using a low-pass filter. Since the sampling device captures M sequences $y'[n]$ at rate f_s , the qM rate f_p sequences $y[nq]$ of the left-hand side of (6.33) can be obtained in digital domain directly. First, each sequence is multiplied by $\exp(-j2\pi k f_p n T_s)$ in order to alias all information into \mathcal{F}_p . Afterwards, these sequences are filtered with a low-pass filter with cutoff at f_p (this corresponds to π/q in digital domain) in order to leave only the frequencies inside \mathcal{F}_p . Finally, the sequences are decimated by q in order to obtain the f_p -rate sequences appearing in (6.33).

In principle, any M -channel MWC device can be collapsed to a device with only one channel having $f_s = M \cdot f_p$ (M odd). However, as the authors note in [64], this approach requires M digital filters and as q grows, the digital filter requires more taps. This increases the computational load.

6.2 Evaluation of the MWC

The MWC is the only of the three presented sampling methods where a working prototype was presented. The authors implemented the MWC for a wideband multiband signal

with $K = 6$ bands with a maximum bandwidth of $B = 19$ MHz at the Nyquist rate of 2.075 GHz [71]. The choice of $f_p = 19.212$ MHz translates into $N = 108$ spectrum slices. However, the number of channels has been reduced to $M = 4$ channels by using the method described in section 6.1.5 with $q = 3$, requiring a sampling rate of at least $f_s = 3f_p = 57.64$ MHz per channel. In the prototype 70 MHz ADCs have been used such that the overall sampling rate is $f_{\text{tot}} = 4f_s = 280$ MHz instead of $f_N = 2.075$ GHz. This means a saving of a factor 7 compared to a conventional Nyquist ADC.

6.2.1 The MWC and Non-Blind sampling

Provided that \mathcal{S} is known, the device works completely without any CS ingredients. Given the sample sequences $\mathbf{y}[n]$, (6.26) can be solved easily with conventional technology within the digital signal processor (DSP): It consists solely of linear matrix operations. These operations only depend on the current sample (i.e., it is memory-less) and a small size, constant matrix Φ . Thus, the signal can be acquired and processed in real time in this case. This also is visualized in figure 6.4. The sampling device is based on well established technology since it only requires demodulation of spectrum slices and conventional ADCs.

Suppose a multiband signal with K transmissions where each band has a bandwidth of B . With knowledge of the center frequencies f_i , each band can be demodulated to the baseband and sampled at rate B such that the overall sampling rate is given by $f_{\text{tot}} = KB$ (other methods include non periodic nonuniform sampling, for example, [72]). Landau showed that the minimum sampling rate in such a scenario is $f_{\text{tot}} = KB$ [73]. This is also true for the MWC with known \mathcal{S} as long as $M \geq K$ because in this case (6.26) is left invertible.

From this point of view, the MWC can be regarded as an extension where the center frequencies f_i are *not* known. The sampling device only needs an additional detector for the f_i which is nothing more than the CTF. For unknown spectral support, the minimum sampling was shown to be $f_{\text{tot}} = 2KB$ [64]. In this case, $\mathbf{z}(f)$ is unique for every $x(t)$ and can be solved only theoretically with (6.5). The increased value of (6.24) stems from the fact that the support must be found using a finite program. For this reason, the action of the MWC can be reduced to the CTF: As soon as \mathcal{S} is found it is clear that the MWC will reconstruct the signal. Experiments have shown that the CTF always reconstructs the spectral support successfully given that the discussed requirements on \mathbf{A} and M are met.

6.2.2 Number of channels

From an analytic point of view, the number of required channels is given by (6.23). However, it is not as easy as for the RD to experimentally verify this due to the “real” analog behavior of the MWC. In general, the experiment needs to place K bands into

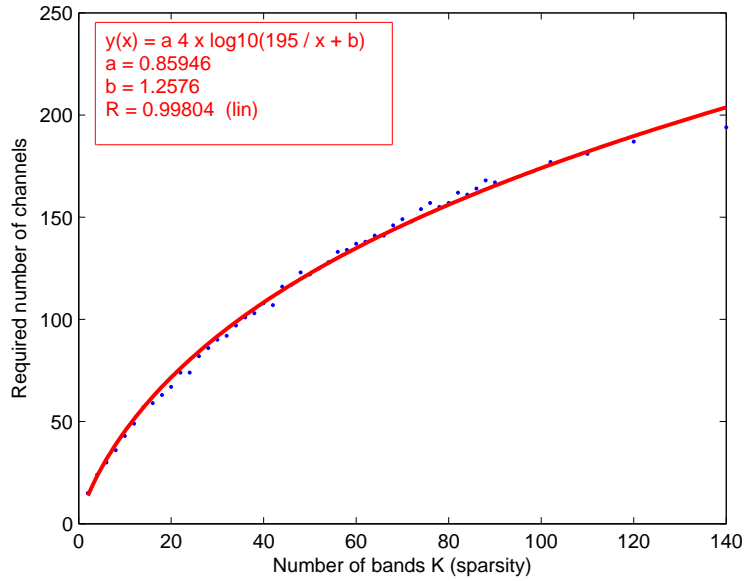


Figure 6.6: Required number of channels for a given number of bands K . The result is similar as predicted by (6.23) and similar as for other methods involving CS such as the RD in (4.21) (code)

N slices and determine the number of channels M for successful recovery of \mathcal{S} . A first, naïve approach places some compactly supported, discrete data into K spectrum slices. However, this is not a realistic scenario. Therefore, the same approach as in [64] has been used, where K bands were modulated onto their respective carrier frequencies f_i by using the formula

$$x(t) = \sum_{i=1}^{K/2} \text{sinc}(B(t - \tau_i)) \cos(2\pi f_i(t - \tau_i)) \quad (6.35)$$

which in theory corresponds to rectangular bands of width B (the τ_i are optional time shifts in order to visually separate the bands in time domain). However, due to the finite representation of the sinc function, each band does not occupy $2K$ slices (at maximum) but contains few energy in other slices as well. For small K this is not an issue, however, for large K , the CTF recovers a larger support set.

The general approach is to draw the $K/2$ carrier frequencies within $[0, f_N/2]$ at random. However, with $B \approx f_p$ chances are high that two transmissions fall into the same spectrum slice as K grows which distorts the result. Therefore, in order to get reproducible results,

the carrier frequencies have been drawn from the following set, such that each possible f_i is centered in the corresponding spectrum slice:

$$f_i \in \{n \cdot f_p \mid n = 1, \dots, N_0\} \quad (6.36)$$

The result of the experiment is shown in figure 6.6. For the experiment $N = 195$ spectrum slices and $f_N = 10$ GHz have been used. The input signal was built according to (6.35) with $B = 0.9 \cdot f_p$ and the carriers have been drawn from (6.36). For each pair of (N, K) , the smallest number M was sought which reconstructed at least 95 out of 100 trials. The CTF was implemented using an MMV variant of the orthogonal matching pursuit (OMP) algorithm obtained from [74]. The logarithmic fit shows a similar behavior as the sampling rate of the RD (4.21) and coincide with (6.23). The missing factor of 2 in the denominator inside the logarithm stems from the fact that the model (6.36) does not enable a band to share more than one spectrum slice. Although the experiment confirms (6.23), only a real hardware implementation can prove the suitability.

While the number of channels might be reduced with the method described in section 6.1.5, the hardware layout of the device still depends on the input signal (the number of bands). However, in contrast to the finite rate of innovation (FRI) method it easily is possible to design an MWC for a maximum of L transmissions which works happily for any number of $K < L$ transmissions.

6.2.3 Non-ideal low-pass filter

The role of the low-pass filter appears implicitly in (6.18). Its task is to leave the spectral data for $f \in \mathcal{F}_s$ only. When $h(t)$ is not an ideal rectangular filter, the vector $\mathbf{z}(f)$ becomes distorted and $\hat{x}(t)$ from (6.27) will be distorted as well. Figure 6.7 shows an experiment using a Butterworth filter with order 6 instead of an ideal low-pass filter. The distortion is especially problematic when a band shares contributions in two spectrum slices. Figure 6.7(b) shows that the first and the third transmission have high distortion in the middle of the spectra. Depending on the modulation method of the transmission this might not be a problem and data still might be recovered successfully. When one band shares only a small part in a specific spectrum slice, the CTF recovers a different support set. For example, the second band is distorted in such a way that the spectrum slice at 2.2 GHz is not regarded as part of the support set. There might be room for improvements by choosing other filter types or by slight modification of the cutoff frequency.

Since the usage of non-ideal filters is an issue encountered in classic sampling theory, a simple countermeasure is to oversample the input signal and append a digital low-pass with a sharp transition. In [75], it was shown analytically that all filters satisfying Nyquist's inter symbol interference (ISI) criterion are able to recover a multiband signal with the MWC perfectly. In particular, this includes Kaiser windows and filters which are readily available in hardware such as raised-cosine filters. Additionally, it is even

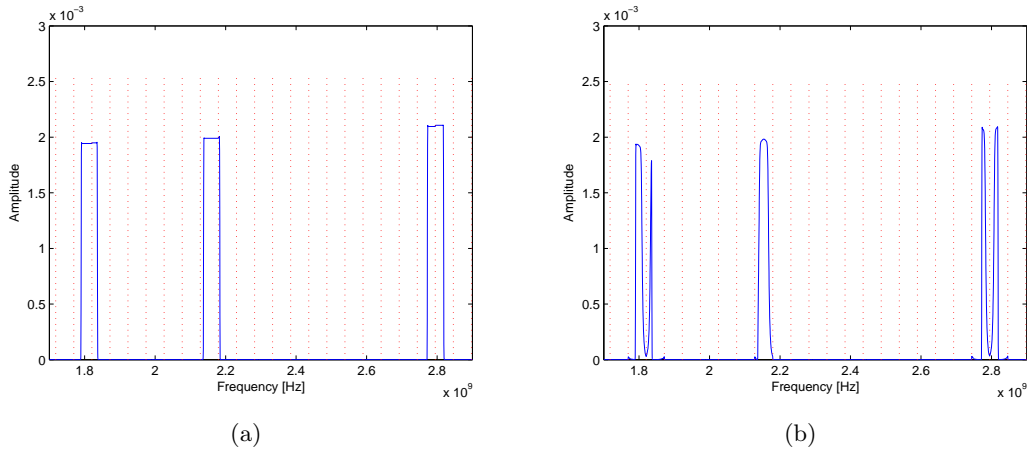


Figure 6.7: Effect of using non-ideal low-pass filters (a) ideal low-pass filter (b) Butterworth filter of order 6. Distortion is especially problematic when a band shares contribution within two spectrum slices (code)

possible to achieve almost perfect reconstruction by digital compensation. In [75], the authors showed how to compensate a typical Butterworth filter successfully by using slight oversampling and additional processing in digital domain.

The prototype board from [71] uses a 6-order elliptic filter and an oversampling factor of 1.21 followed by this digital compensation.

6.2.4 Non-ideal modulation sequences

An important question is if the MWC can deal with non-ideal modulation sequences $p_i(t)$. The fundamental property on $p_i(t)$ is the periodicity with f_p since this supports the required aliasing. Other properties are not so important, however, the Fourier coefficients must at least resemble the coefficients in the matrix Φ . A first experiment tested whether the MWC works with non-sharp transitions the same way as for the RD (cf. section 4.3.5) or not. For the experiment a high resolution signal was created from low-rate sequences $p_i[n]$ at Nyquist rate using the MATLAB command `resample`. The signal shape of $p_i(t)$ looks the same as in figure 4.12. The MWC recovered the support as well as the signal $\hat{x}(t)$ without any problems. While the $p_i(t)$ do not have any sharp transitions in this configuration, the result is not surprising because the Fourier coefficients effectively stay the same.

Another experiment observed the effect of random jitter (the same way as described in section 4.3.5). Already small values of σ^2 renders the output of the MWC useless: The CTF as well as reconstruction via (6.26) fail because in this configuration the Fourier coefficients of $p_i(t)$ do not resemble the entries in Φ anymore.

However, the authors argue that this is not a major problem because the entries in Φ can be calibrated after manufacturing [63]. In the prototype [71] the authors show that this calibration indeed is possible. It shows a practical example of a sequence $p_i(t)$ alternating at 2 GHz whose pulse shape is far from perfect: There are neither sharp transitions nor alterations exactly at the Nyquist rate.

6.2.5 Sub-Nyquist processing

An important feature of a sub-Nyquist sampling system is the ability to process the samples digitally at a rate below the Nyquist rate. In the absence of this feature, the digital reconstruction needs to recover a digital equivalent $x[n]$ of $x(t)$ at the Nyquist rate which might be problematic in terms of computational power or memory, especially with high-rate signals. While the RD lacks this feature, the FRI framework offers this feature by definition: It reconstructs time delays and amplitudes and therefore provides encoded information directly.

For the MWC the information is contained within the K bands, it may be encoded by arbitrary modulation. In the context of MWC, baseband processing means that it is possible to obtain the low-rate samples of a single band without reconstructing $x(t)$ at the Nyquist rate. The MWC reconstructs $\mathbf{z}_S[n]$ (6.26) where the DSP has direct access to the samples of a specific spectrum slice containing some signal energy where the samples are reconstructed at a low rate f_p . However, there are two problems: First, the exact carrier frequencies f_i are not known (only the indexes of the spectrum slices containing signal energy) and second, a band might share contributions over two spectrum slices.

In [76], the authors proposed an extension to the MWC which supports to recover the samples of the bands at a low rate. The method assumes that each band is modulated using a general quadrature modulation, that is, $x(t)$ consists of $K/2$ transmissions of the form:

$$x(t) = \sum_{i=1}^{K/2} I_i(t) \cos(2\pi f_i t) + Q_i(t) \sin(2\pi f_i t) \quad (6.37)$$

The signals $I_i(t)$ and $Q_i(t)$ directly encode the information. Additionally for the method to work, the minimum width of a band and the minimum spacing between two bands must be known.

6.2.6 Real-time processing

As discussed above, the MWC offers real-time processing once the support is given. The reconstructed output of (6.26) is only dependent on the current input samples. In contrast to the RD, reconstruction does not work block-wise but rather continuously. Only the

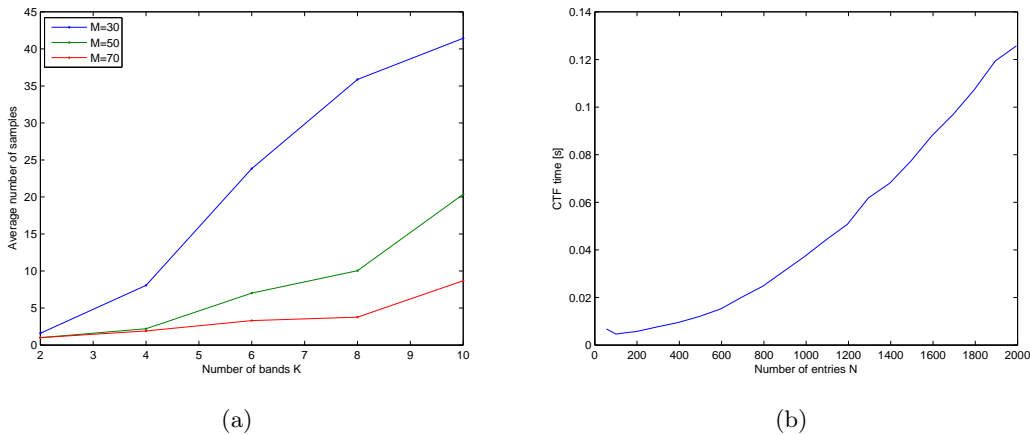


Figure 6.8: Real-time capability of the MWC (a) Average number of samples required for the CTF (i.e., for (6.25)) in order for the support recovery to succeed. This number is dependent on the number of channels and is quasi-linear by the number of bands K . In general, this number is very small (code) (b) The running time of the CTF is dependent of the number of spectrum slices N and negligible small in a typical scenario (code)

CTF introduces a certain delay: First a few samples have to be collected in order to obtain \mathcal{S} and second, the MMV system (6.9) needs to be solved which requires computation time. As described in section 6.1.4, the number of required samples for the CTF is theoretically small. Figure 6.8(a) shows an experiment how much samples are required for (6.25) in order to obtain enough linearly independent vectors. In the experiment, a multiband signal with $f_N = 10$ GHz and different numbers of bands $K = \{2, 4, 6, 8, 10\}$ was created. The MWC used $f_p = f_s = 51.3$ MHz and the number of channels varied with $M = \{30, 50, 70\}$. It was repeated for 100 trials and the average number of required channels M in case of a successful support recovery was measured.

Figure 6.8(a) verifies that the number of required samples is typically small. For example, a sampling device with $M = 50$ channels requires about 5 samples in each channel only in order to detect the support of a multiband signal with $K = 6$ bands. The number of required samples is quasi-linear in the number of bands K . Furthermore, a sampling device with more channels requires fewer samples. This is intuitive since in this case the vector $\mathbf{y}[n]$ contains more information for every n .

Figure 6.8(b) shows the computation time for the CTF on a typical desktop PC for a constant sparsity of $K = 6$ and different sizes N of $\mathbf{z}(f)$. The important thing is that the computation time is only dependent on the number N since this number defines the size of the MMV system (6.9). A typical application partitions the spectrum only into a few spectrum slices. For example, the prototype in [71] uses $N = 108$ spectrum slices.

The computational load is negligible small in this case.

Another aspect is the change of the support. Under the assumption that the senders will not change their carrier frequencies once chosen, the CTF only needs to find the support at the beginning. However, it is possible without any problems that the transmission continuously change their carrier frequencies. In this case an additional block might observe the signal support and trigger the CTF when the support changes over a specific threshold. When the MWC continuously buffers a certain amount of samples for the MWC there is even no delay in this case. [63] offers a high level view on this method.

6.3 Conclusion

At the first glance, the MWC is the perfect sub-Nyquist sampling device. It avoids any discretization of the signal model as with the RD and uses a real analog sparsity model. It provides any properties which makes it possible to build the device in hardware: The low-pass filters as well as the modulation sequences do not need to be perfect and can be compensated in digital domain. The computational load is very light and suitable for mobile applications. An optional extension provides direct access to the low-rate sequences of the bands without needing to recover the input signal at the Nyquist rate. According to [71], the current barrier are the high frequency modulators since current high-frequency modulators only are suited for sinusoidal modulations. The chipping sequences itself are not an issue: Binary pseudo random number generators alternating at 80 GHz have been reported [77]. Additionally, a prototype shows the feasibility of the system in practice. However, the behavior in real-world scenarios is not yet determined.

On closer examination it turns out that the MWC is merely a classical sampling device which is extended by a spectral-support detector. It does not support to sample real ultra-wide band (UWB) signals since those signals do not consist of separable, narrow bands but rather of pulses compromising low-energy wide-bands. For proprietary communication devices it might be easier to use a low-rate protocol distributing the carrier frequencies used. When using the MWC together with other transmissions, each band must be decoded in order to get access to the own data.

A typical application of the MWC is cognitive radio which is also mentioned by the authors [78]. Another application directly related is spectral sensing where the MWC can be used to detect free spectrum slices for own transmissions. However, in this configuration it suffices to use only the CTF part, degrading the sampling device to a spectrum sensing device.

A disadvantage about the MWC is that the hardware branches directly depend on the number of transmissions and the complexity of the device quickly increases. Furthermore, the allocation of the spectrum slices is rather static and given by the Nyquist rate and a fixed number of cells N . This results in an inflexible signal model.

Chapter 7

Comparison and summary

Table 7.1 shows a table comparing the three sub-Nyquist sampling systems covered by this thesis. Since the methods have different aims, they all use their own type of sparsity. Therefore it is not easy to compare them since the only common goal is to sample below the Nyquist rate. For this reason this chapter will focus on practicability and provides examples for applications. The first section summarizes the most important properties and the following section compares the methods based on the evaluation in this thesis.

7.1 Properties

This section summarizes the most important features of the three discussed techniques given in the upper part of table 7.1.

Input signal and sparsity Each system uses a different type of input signal which defines a different type of sparsity in its signal representation. The random demodulator (RD) deals with signals consisting of pure sinusoids. The sparsity is defined as K out of N possible frequencies.

The finite rate of innovation (FRI) framework published so far only deals with Diracs (or signals which can be reduced to Diracs) where the number of Diracs defines the sparsity. The underlying signal model is real analog.

The modulated wideband converter (MWC) deals with signals consisting of simultaneous transmissions on different carrier frequencies, also referred as multiband signals. The sparsity is defined as the number K out of N possible bands. The MWC is able to decode all K bands simultaneously without knowing their carrier frequencies.

Frameworks and randomness Currently there are two dominant sub-Nyquist sampling frameworks: compressive sampling (CS) and the FRI framework. While some relations between these two frameworks have been established [79], a unifying theory is still missing.

The RD and the MWC use the CS framework which is based on solving underdetermined systems of equations. In both cases, the sampling rates are linear in the sparsity K and include a logarithmic term. For practical implementations, CS requires random linear

	RD	FRI	MWC
input $x(t)$	multitone	pulses	multiband
sparsity	nr. of tones	nr. of pulses	nr. of bands
framework	CS	FRI	classical + CS
randomness	yes	no	yes
prototype	partially	no	yes
sub-Nyquist proc. components	no	yes	yes
	LFSR, modulator, filter, ADC	filter, ADC	LFSR, splitter, M modulators, M filters, M ADCs
comp. load	very heavy	light	very light
mem. requirements	very high	low	very low
typical bandwidth	< 10 kHz	up to ∞	< 80 GHz
real-time proc.	no	yes	yes
stability	good	bad	very good
applications	single sinusoids	UWB, time delay est.	cognitive radio spectrum sensing
component sensitivity	high	moderate	low
practical efficiency barrier	low computation	high floating point precision	moderate modulator
practicability	no	no	yes

Table 7.1: Comparison of the sub-Nyquist sampling system covered by this thesis

combinations of the measurements. Therefore, both the RD and the MWC use random chipping sequences.

By contrast, the FRI framework analytically relates parameters of the input signal to an innovation signal from which the parameters are estimated. The framework requires no randomness. Theoretically the sampling rate is almost at the minimum (e.g., $f_s = 2K + 1$ for $2K$ degrees of freedom). In practice, it is still linear with a small factor in K .

Prototypes For FRI no prototype has been published. Several practical aspects for the RD have been published [42, 33, 80] but no comprehensive prototype which targets real-world applications. The MWC is the only device for which a prototype has been published [71] and which works for real-world applications.

Sub-Nyquist processing The term *sub-Nyquist processing* refers to the ability of the sampling device to extract the encoded information directly at a low rate (at least below the Nyquist rate). In absence of this feature, a digital signal representation at the Nyquist rate has to be reconstructed. This might be problematic in terms of computational power or memory requirements. The RD lacks this feature: The recovery stage requires to reconstruct the signal first at Nyquist rate from which the information (e.g., included frequencies) can be derived afterwards.

The FRI provides this feature by definition: The reconstruction process recovers time delays and weights of pulses which directly provide the encoded information at the information rate.

For the MWC, an extension [76] provides samples (under certain assumptions) at the rate of the bandwidth of the different bands. Although this method does not reach the information rate, it is about N times lower than the Nyquist rate (e.g., $N = 195$ for a practical wideband scenario). This method works when the transmissions can be represented in a general quadrature modulation form (6.37).

Components Certainly all methods require a (low-rate) analog-to-digital converter (ADC) in order to convert the analog signal into a digital one. Additionally all methods require preprocessing in terms of an analog filter which is a low-pass filter in most cases. The methods relying on random sequences (RD and MWC) additionally require both a signal generator (which can be a linear feedback shift register (LFSR) in both cases) and a modulator or mixer (which mixes the input signal with the random sequences).

The MWC requires all components (except the splitter and the LFSR) M times (one for each physical channel).

7.2 Summary

The lower part of table 7.1 summarizes the evaluation in this thesis.

Computational load The computational load for the RD is very heavy: Since the reconstruction recovers the signal at the Nyquist rate, the optimization problem has the cardinality of the Nyquist rate. When using spectral compressive sampling (SCS), the computational load is even higher. This makes it impossible to use the RD for high frequency or wideband signals.

In the absence of noise, the computational load for the FRI framework is directly related to the number of pulses. In the presence of noise, the recovery process needs to perform a singular value decomposition (SVD) on a matrix whose dimensions are approximately the number of samples, followed by few matrix operations and an eigenvalue decomposition. These operations are very fast on a modern computer (less than a second for a 1000×1000

matrix). Depending on the method used, a fast Fourier transform (FFT) of the length of the samples might be necessary. It also can be implemented very fast.

The computational load of the MWC is very light. Once the support is recovered, only linear, low-dimensional (in the range of K and M) matrix operations are required. The continuous-to-finite converter (CTF) is only needed when the signal support changes. The complexity is defined by the number of channels M and the number of spectrum slices N . The running time is negligible for a typical application.

Memory requirements The huge memory requirements are some of the fundamental problems of the RD. For a GHz-signal the memory requirements are several hundred TB. Since each entry has to be accessed at least a few times, the RD is not suitable for high rates in the near future.

The matrix operations for the FRI reconstruction require memory which is quadratically in the number of samples and typically low. Thus, the memory requirements for the FRI framework are low.

The MWC has the least memory requirements: The CS matrix has only the dimensions $M \times N$ and the matrices in the multiple measurement vector (MMV) setup (6.9) are about $M \times 2K$.

Typical bandwidth The high computational and memory requirements for the RD restrict the usage to very low signals, for a typical PC even less than 10 kHz.

The FRI framework deals with signals which generally have a high bandwidth. In theory, the bandwidth is even ∞ because Diracs (or piecewise polynomials) have an infinite bandwidth. The trick is to project the signals to a finite-dimensional subspace (e.g., with a low-pass filter to a low-pass approximation) which contains all information of the input signal $x(t)$.

The MWC samples classical multiband signals. The technological barrier is the hardware. Under the assumption of a 80 GHz LFSR [77] the practically achievable bandwidth is at least a few GHz.

Real-time processing Real-time processing means that the sampling device is able to process data in real time while sampling. The RD needs to collect samples for a specific amount of time (e.g., 1 s for a 1 kHz signal) prior to reconstruction. The digital reconstruction takes additional time. Therefore the RD lacks this feature.

The FRI framework as well as the MWC provide real-time processing. With a small support kernel in FRI it is possible to have quasi-local reconstruction.

The reconstructed output of the MWC only depends on the current input sample. Therefore the MWC reconstructs in real time once the support is given. Even including the CTF, the device operates in quasi real time because very few samples are required

(in the region of $2K$). This delay can be further decreased by buffering samples for the CTF continuously.

Stability In this instance, stability means that the sampling device works properly when the number K increases. This is true for the RD as well as for the MWC.

The FRI framework is stable when the innovation signal is the Fourier transform. However, this only works reliable for periodic and therefore infinite signals. The FRI framework based on local kernels is very unstable until now when used with more than $K \approx 5$ Diracs.

Applications A typical application for the RD is the sampling of signals consisting of K pure, arbitrary sinusoids. It is important to note that the RD can not be used to sample signals such as in orthogonal frequency-division multiplexing (OFDM) although the signal definition looks similar as (4.7) [81]. The reason is that the RD requires the spectrum to be sparse and therefore the sinusoids have a long extension in time. The sinusoids in OFDM by contrast have a very short extension of $1/\Delta$ (with Δ being the carrier spacing) resulting in a very dense spectrum. Due to the high computational load, the RD can be used for low-rate signals only, for which cheap commercial ADCs exist [5]. For the reasons mentioned above, there is no real-world application for the RD.

A typical application for the FRI framework is ultra-wide band (UWB) and in particular time-delay estimation of pulses. The time-delays can modulate information for UWB. The big advantage is that an UWB receiver based on FRI could operate at the information rate rather than at the high chip rate [48].

Applications for the MWC are cognitive radio and spectrum sensing.

Component sensitivity The term *component sensitivity* covers the question whether all components are physically realizable and if yes, if the sampling system is sensitive to slight deviations. A real ADC has an property termed *analog input bandwidth* [82] which follows from the implicit assumption that the sampling obeys the Whittaker-Kotelnikov-Shannon (WKS) theorem anyway. The meaning of this property is that the bandwidth of the input signal $y(t)$ for the low-rate ADCs should not exceed roughly twice the sampling rate of the ADC. In particular, $y(t)$ must not be at the Nyquist rate of $x(t)$. In practice this is avoided by an appropriate low-pass filter which matches the sampling rate of the ADC. This is true for the RD since the analog filter $h(t)$ is an integrator whose cutoff frequency matches the sampling rate f_s . Also the MWC and the FRI framework for periodic signals use a low-pass filter matching the sampling rate. Only the polynomial spline kernels produce an ADC input signal $y(t)$ with a theoretically infinite bandwidth which might cause aliasing in practice.

The RD uses an integrator which is easy to realize where a small bias is not an issue. The MWC theoretically needs an ideal low-pass filter, however, physically realizable

filters such as raised-cosine filters are possible. Furthermore, any real low-pass filter can be compensated in digital domain. The FRI framework for periodic signals also requires an ideal low-pass filter. However, with an oversampling factor higher than 20, ordinary Butterworth filters work. While spline kernels might be difficult to realize, exponential splines can be reduced to rational transfer functions which include Butterworth filters by definition. However, since high-order filters are difficult to realize, this method is restricted to small size kernels and therefore a small number of pulses. For example, a 9-th order Butterworth filter can sample about 4 Pulses only.

Chipping sequences are required by the RD and the MWC. For the RD, the chipping sequences must alternate exactly at the Nyquist rate. In the presence of non-ideal sequences, the performance of the RD decreases very fast. The MWC works as long as the first N Fourier coefficients approximately match the entries in the CS matrix. Therefore the edges do not need to be perfectly sharp. Clock jitter prevents the MWC from working. However, since the only requirement is that the Fourier coefficients of the chipping sequences match the entries in the CS matrix, the entries can be calibrated for imperfect sequences.

Practical efficiency The quotient $\eta = f_N/f_{\text{tot}}$ describes how efficient the sub-Nyquist sampling system works in practice in terms of the sampling rate. This is an important property for a sub-Nyquist sampling system. For the RD, f_{tot} is theoretically linear in K and logarithmically in the Nyquist rate. However, due to model assumptions, f_s needs to divide f_N which provides good savings only for very small values of K . When using SCS, η is decreased even further by at least a factor of 4. Assuming a constant sparsity K , η increases as the Nyquist rate increases. For example, assume an input signal consisting of only $K = 5$ frequencies and a Nyquist rate of 2048 Hz. This translates to an utilization of 0.25% only. Using spectral iterative hard thresholding (SIHT) via root multiple signal classification (MUSIC) requires a sampling rate of $f_s = 154$ Hz where the next number which divides 2048 is 256. This results in an efficiency of $\eta = 8$. However, due to the high computational requirements of the reconstruction algorithm, the Nyquist rate is limited to a few kHz.

Since the bandwidth of FRI signals is theoretically infinite and the sampling rate depends only on the degrees of freedom of the signal, the efficiency is very good and tends to infinity: $\eta \rightarrow \infty$.

The total sampling rate of the MWC depends linearly on the maximum width of the bands B , the number of bands K and logarithmically in the Nyquist rate. The efficiency increases as the number of spectrum slices increases. However, the more spectrum slices, the more channels are required. A reduction of channels in turn increases the total sampling rate to the same factor as the number of channels. In practice, the sampling rates must be increased in order for the digital compensations to work. For a practical setup such as in the prototype of the MWC [71], the efficiency is only about $\eta = 7$.

Technological barrier This point lists the most fundamental technological barriers for each system. For the RD, the technological barriers undoubtedly are the high memory and computational requirements which extend today's capabilities by magnitudes.

The most fundamental problem in the FRI framework is the inherent numerical instability. The system works perfectly, if it is described in an analytical way only. However, as soon as the actions of the framework need to be implemented using finite precision, the reconstruction becomes unstable quickly due to the high numbers involved. The current FRI framework only works with the hypothetical assumption of infinite floating point precision.

From a theoretic point of view, the MWC does not have any limitations. The technological barriers are given by the hardware, in particular by the sign alternating functions and the modulators.

Practicability The practicability addresses the question whether the sampling devices can be built in practice and if it makes sense to build them. The discussion so far gives an answer to this question. While it is possible to build the RD in practice, it is very sensitive to model mismatch. Since it does not provide notable advantages compared to a Nyquist ADC, it does not make sense to use the RD in practice.

The FRI framework for periodic signals works well under certain assumptions. However, since it only applies to periodic signals the practical usage is limited. With the usage of compactly supported kernels, the FRI framework would be able to sample high-rate UWB signals. However, the inherent stability problems limit the practical value.

The MWC is the one of the three presented sampling systems which works and can be built in practice. However, this system has certain limitations which make it questionable if the MWC is advantageous over conventional systems.

Chapter 8

Conclusion

This thesis studied three sub-Nyquist sampling methods: The random demodulator (RD), the finite rate of innovation (FRI) framework and the modulated wideband converter (MWC).

The RD directly applies knowledge from compressive sampling (CS) framework to analog signals. This results in an easy implementation of CS but also implies some issues: First, a discrete signal model which is very sensitive to model mismatch. Second, the CS framework requires to reconstruct a full-size vector of a large, but sparse signal model digitally rather than parts of it. For the RD, this size directly depends on the Nyquist rate. Not least, this is the reason for the extraordinary high computational requirements. To conclude, the RD shows that direct application of the discrete CS framework does not work in practice. However, CS works well for discrete signals.

The FRI framework is the most interesting concept. The recovery procedure is light and it directly reveals signal parameters. This avoids further complex processing steps in many cases. A typical application which would instantly profit from a working FRI framework is ultra-wide band (UWB). Unfortunately, the promising local reconstruction method is too unstable and too sensitive for practical applications. An extension [61] promises stable reconstruction for real-world finite signals. However, the capability of application to real-world signals yet needs to be determined.

The MWC is a multichannel sampling device which is based on mirroring parts of the spectrum to the baseband. Important concepts are based on well established methods such as demodulation while borrowing a key-concept for reconstruction from the discrete CS framework. The integral part, the continuous-to-finite converter (CTF), reduces a real sparse, analog signal model to a finite dimensional problem which can be solved easily. Additionally, a prototype shows that practical implementation is possible. The main drawbacks about the MWC are the static allocation of spectrum cells and the dependency of the hardware layout from signal parameters. Additionally, it is to be determined yet how the MWC behaves in real-world scenarios.

While the well-established Whittaker-Kotelnikov-Shannon (WKS) theorem is linear, the reconstruction of sparse analog signals is non-linear. For all of the three methods, this inherent non-linearity is problematic since deviations caused by model mismatch or non-ideal components lead to huge deviations of the reconstructed signal. By contrast,

the WKS theorem is linear and simple. First, non-ideal components are not an issue: Non-ideal low-pass filters such as Butterworth filters or track-and-hold instead of sample-and-hold work well in practice. Second, model mismatch is not an issue: The anti-aliasing filter forces the input signal into the required, linear signal model. Finally, low-pass approximations of signals not fitting the signal model often are acceptable in practice. Developing a sampling device which provides similar flexibility in practice is a challenging task.

In general, the idea to trade sampling rate with additional computational power is not yet achievable, where the MWC is the most likely of the three concepts to be able to achieve this goal. In case of the RD, the computational requirements are too high, by order of magnitudes, compared to the computational power which would be available when using low-rate analog-to-digital converters (ADCs). Generally, the algorithms are yet too immature. For the moment, digitally assisted ADCs seem to be the more apparent solution to the ADC bottleneck. However, the research around sampling of sparse analog signals below the Nyquist rate is very dynamic.

Further research The FRI framework provides the most interesting concept. However, currently used methods either use too unrealistic assumptions or are too unstable. Finding a method which combines the advantages of sampling of periodic Diracs with local sampling would solve the issues with FRI.

Another interesting concept is the reduction of a real, analog sparse signal model to a finite dimensional problem which can be tracked in practice. The MWC makes use of this strategy. However, the underlying sparsity model is valid for multiband signals only. A further step is to extend this method to more general types of sparsity. A first framework termed CS for analog signals has been established [20, 83].

From a practical point of view, a further step is to evaluate the MWC in depth and to determine the feasibility to real-world scenarios where the focus is cognitive radio. For this evaluation, the device needs to be built in hardware and tested with real-world communication signals. In particular, the robustness to noise, calibration with non-ideal filters and chipping sequences need further investigation. A practical evaluation is especially important for the baseband processing system since the function of this system is an integral part for real-world scenarios where multiple (extrinsic) signals need to be decoded simultaneously.

Appendix A

Derivation of the coefficients in the polynomial reproduction formula

The derivation starts with (5.40). For the derivation I set $\varphi = \beta_N$. The coefficients $c_{m,n}$ can be obtained using the dual of φ , $\tilde{\varphi}$ [30]:

$$c_{m,n} = \int_{-\infty}^{\infty} t^n \tilde{\varphi}(t - m) dt \quad (\text{A.1})$$

However, even if the dual would be known, solving the infinite integral only is feasible when the dual has finite support. This is the case for the B-spline itself but not for its dual. A closer look at (A.1) tells that this is nothing more than a convolution (under the assumption that $\tilde{\varphi}$ is symmetric which is the case):

$$c_{m,n} = \int_{-\infty}^{\infty} t^n \tilde{\varphi}(-(m - t)) dt = \int_{-\infty}^{\infty} t^n \tilde{\varphi}(m - t) dt = (t^n * \tilde{\varphi})(m) \quad (\text{A.2})$$

Now, this can be transformed to Fourier domain:

$$(t^n * \tilde{\varphi})(m) = \mathcal{F}^{-1} \left\{ \mathcal{F} \{t^n\} \tilde{\Phi}(\omega) \right\} \quad (\text{A.3})$$

$$= \mathcal{F}^{-1} \left\{ j^n \sqrt{2\pi} \delta^{(n)}(\omega) \tilde{\Phi}(\omega) \right\} \quad (\text{A.4})$$

$$= j^n \sqrt{2\pi} \mathcal{F}^{-1} \left\{ \delta^{(n)}(\omega) \tilde{\Phi}(\omega) \right\} \quad (\text{A.5})$$

Writing the inverse of this expression yields:

$$j^n \sqrt{2\pi} \frac{1}{\sqrt{2\pi}} \int_{-\pi}^{\pi} \delta^{(n)}(\omega) \tilde{\Phi}(\omega) e^{j\omega m} d\omega = j^n \int_{-\infty}^{\infty} \delta^{(n)}(\omega) \underbrace{\tilde{\Phi}(\omega) e^{j\omega m}}_{f(\omega)} d\omega \quad (\text{A.6})$$

It is known that [84]:

$$\int_{-\infty}^{\infty} \delta^{(n)}(x) f(x) dx = (-1)^n f^{(n)}(0) \quad (\text{A.7})$$

so that (A.6) becomes

$$j^n \int_{-\infty}^{\infty} \delta^{(m)}(\omega) f(\omega) d\omega = j^n (-1)^n \left. \frac{\partial^n}{\partial \omega^n} f(\omega) \right|_{\omega=0} \quad (\text{A.8})$$

Now the whole procedure has been reduced to calculate the derivative of $f(\omega)$ and set the result to zero. An open question is how to obtain the dual of φ . Since the reproduction formula spans a vector space, φ must be at least bi-orthogonal to $\tilde{\varphi}$. This translates in Fourier domain to [85]:

$$\tilde{\Phi}(\omega) = \frac{\Phi(\omega)}{\sum_{k \in \mathbb{Z}} |\Phi(\omega + 2\pi k)|^2} \quad (\text{A.9})$$

The Fourier transform of a B-Spline of order N is [58]:

$$B_N(\omega) = \Phi(\omega) = \left(\frac{\sin(\omega/2)}{\omega/2} \right)^{N+1} = \text{sinc}^{N+1}(\omega/2) \quad (\text{A.10})$$

The first step is to calculate the denominator of (A.9). The following derivation of the sum is borrowed from [86]. In order for this derivation to work, I set $L = N + 1$ temporarily:

$$\sum_{k \in \mathbb{Z}} |\Phi(\omega + 2\pi k)|^2 = \sum_{k \in \mathbb{Z}} \left| \text{sinc} \left(\frac{1}{2}(\omega + 2\pi k) \right) \right|^{2L} = \sum_{k \in \mathbb{Z}} \left| \text{sinc} \left(\frac{1}{2}(\omega + 2\pi k) \right) \right|^{2L} \quad (\text{A.11})$$

and because $2L$ is always even:

$$= \sum_{k \in \mathbb{Z}} \frac{\sin^{2L}(\frac{1}{2}(\omega + 2\pi k))}{(\frac{1}{2}(\omega + 2\pi k))^{2L}} = \sum_{k \in \mathbb{Z}} \frac{\sin^{2L}(\frac{\omega}{2} + \pi k)}{(\frac{\omega}{2} + \pi k)^{2L}} \quad (\text{A.12})$$

Because of the periodicity it is known that

$$\sin^{2L}(x + \pi k) = \sin^{2L}(x) \quad (\text{A.13})$$

such that (A.12) becomes

$$= \sin^{2L} \left(\frac{\omega}{2} \right) \sum_{k \in \mathbb{Z}} \frac{1}{\left(\frac{\omega}{2} + \pi k \right)^{2L}} \quad (\text{A.14})$$

And finally the following relation is used [87]:

$$\sum_{k \in \mathbb{Z}} \frac{1}{(x + \pi k)^{2L}} = -\frac{1}{(2L - 1)!} \frac{d^{2L-1}}{dx^{2L-1}} \cot x \quad (\text{A.15})$$

in order to obtain finally:

$$\sum_{k \in \mathbb{Z}} \left| \operatorname{sinc} \left(\frac{1}{2}(\omega + 2\pi k) \right) \right|^{2L} = -\sin^{2L} \left(\frac{\omega}{2} \right) \frac{1}{(2L - 1)!} \frac{d^{2L-1}}{d \left(\frac{\omega}{2} \right)^{2L-1}} \cot \left(\frac{\omega}{2} \right) \quad (\text{A.16})$$

and again with $L = N + 1$, this becomes:

$$\sum_{k \in \mathbb{Z}} |\Phi(\omega + 2\pi k)|^2 = -\sin^{2(N+1)} \left(\frac{\omega}{2} \right) \frac{1}{(2N + 1)!} \frac{d^{2N+1}}{d \left(\frac{\omega}{2} \right)^{2N+1}} \cot \left(\frac{\omega}{2} \right) \quad (\text{A.17})$$

Therefore, together with the definition of $\Phi(\omega)$ from (A.10), (A.9) becomes:

$$\tilde{\Phi}(\omega) = \frac{(2N + 1)!}{\left(\frac{\omega}{2} \right) \sin \left(\frac{\omega}{2} \right)^{N+1} \frac{d^{2N+1}}{d \left(\frac{\omega}{2} \right)^{2N+1}} \cot \left(\frac{\omega}{2} \right)} \quad (\text{A.18})$$

and finally substituting for $f(\omega)$:

$$f(\omega) = \frac{(2N + 1)!}{\left(\frac{\omega}{2} \right) \sin \left(\frac{\omega}{2} \right)^{N+1} \frac{d^{2N+1}}{d \left(\frac{\omega}{2} \right)^{2N+1}} \cot \left(\frac{\omega}{2} \right)} e^{j\omega m} \quad (\text{A.19})$$

Since this function is not well defined it is better to use the limit:

$$c_{m,n} = j^n \lim_{\omega \rightarrow 0} f(\omega) \quad (\text{A.20})$$

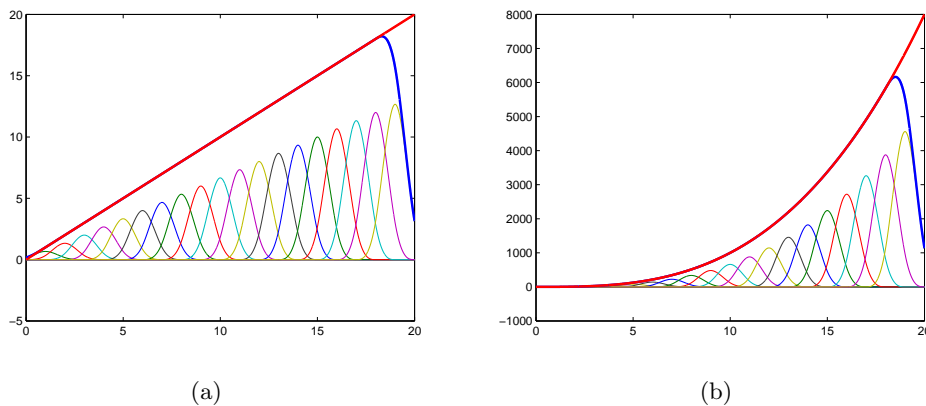


Figure A.1: Reproducing polynomials with B-splines in interval $[0, 20)$ (a) reconstructing the linear polynomial (b) reconstructing a cubic polynomial (code)

Examples for a cubic spline

Equation (A.20) provides closed-form expressions for the reproduction coefficients. For a cubic spline ($N = 3$) the coefficients are given by:

$$\begin{aligned}
 c_{m,0} &= 1 \\
 c_{m,1} &= m \\
 c_{m,2} &= \frac{1}{3}(-1 + 3m^2) \\
 c_{m,3} &= -m + m^3
 \end{aligned} \tag{A.21}$$

Figure A.1 shows how the functions $f(t) = t$ and $f(t) = t^3$ are reproduced using these coefficients.

Appendix B

Derivation of the coefficients in the exponential reproduction formula

Calculating the coefficients $c_{m,n}$ is much more straight forward than for the polynomial case. For the first order case, the coefficients trivially are found by [60]:

$$e^{\alpha t} = \sum_{m \in \mathbb{Z}} e^{\alpha m} \beta_{\alpha}(t - m) \quad (\text{B.1})$$

Starting from the reproduction formula in (5.46), the kernel $\beta_{\mathbf{a}}$ must be able to reproduce the polynomials $e^{(\alpha_0 + \lambda d)t} = e^{\alpha_n t}$ with $n = \{0, 1, \dots, N\}$. From [60] it is known that an arbitrary E-spline $\beta_{\mathbf{a}}$ is produced by convolution of their single elements:

$$\beta_{\mathbf{a}}(t) = (\beta_{\alpha_0} * \beta_{\alpha_1} * \dots * \beta_{\alpha_N})(t) \quad (\text{B.2})$$

Now, take an arbitrary $\alpha_k = \alpha_0 + \lambda k$. Reproducing this polynomial is possible with (B.1). However, it must be possible with the kernel defined in (B.2) also. Therefore, define a kernel $\varphi(t)$ as:

$$\varphi(t) = (\beta_{\alpha_{e_0}} * \beta_{\alpha_{e_1}} * \dots * \beta_{\alpha_{e_{N-1}}})(t) \quad e_i \in \{0, 2, \dots, N\} \setminus \{k\} \quad (\text{B.3})$$

Now (B.1) is convolved with this kernel:

$$\varphi(t) * e^{\alpha_k t} = \varphi(t) * \sum_{m \in \mathbb{Z}} e^{\alpha_k m} \beta_{\alpha_k}(t - m) \quad (\text{B.4})$$

For the left side we have:

$$(\varphi * e^{\alpha_k t})(t) = \int_{-\infty}^{\infty} \varphi(\tau) e^{\alpha_k(t-\tau)} d\tau = e^{\alpha_k t} \int_{-\infty}^{\infty} \varphi(\tau) e^{-\alpha_k \tau} d\tau \quad (\text{B.5})$$

And for the right side we have simply

$$\varphi(t) * \sum_{m \in \mathbb{Z}} e^{\alpha_k m} \beta_{\alpha_k}(t - m) = \sum_{m \in \mathbb{Z}} e^{\alpha_k m} \underbrace{(\varphi * \beta_{\alpha_k})}_{\beta_{\mathbf{a}}}(t - m) \quad (\text{B.6})$$

because of the linearity of the convolution operator. Putting together (B.5) and (B.6) yields

$$e^{\alpha_k t} = \sum_{m \in \mathbb{Z}} \underbrace{\frac{e^{\alpha_k m}}{\int_{-\infty}^{\infty} \varphi(\tau) e^{-\alpha_k \tau} d\tau}}_{c_{m,k}} \beta_{\mathbf{a}}(t - m) \quad (\text{B.7})$$

Since $\varphi(t)$ is an E-spline by definition, it has finite support. Therefore the integral in (B.7) can be solved easily numerically.

Examples

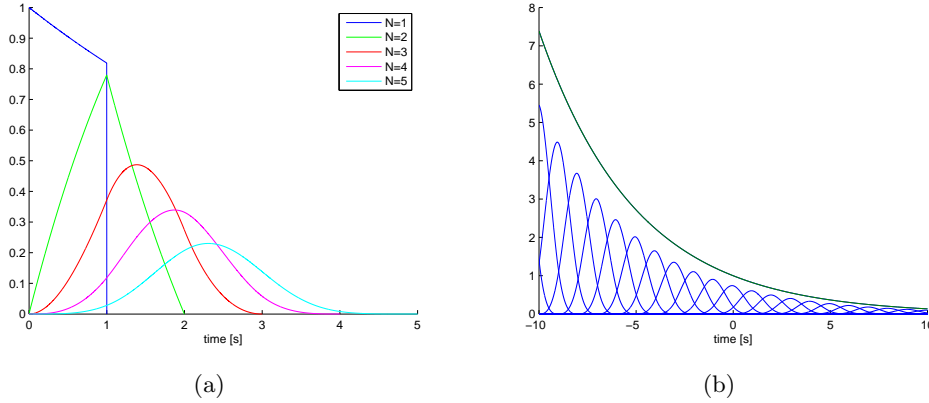


Figure B.1: (a) Family of E-splines with $\alpha_n = -0.2 - 0.1n$ (b) Reproduction of the exponential $e^{-0.2t}$ with E-spline of order $N = 4$

Bibliography

- [1] H. D. Lüke, “The origins of the sampling theorem,” *IEEE Communications Magazine*, vol. 37, no. 4, pp. 106–108, Apr. 1999.
- [2] H. Nyquist, “Certain topics in telegraph transmission theory,” *Proceedings of the IEEE*, vol. 90, no. 2, pp. 280–305, Feb. 2002, (reprint).
- [3] V. A. Kotelnikov, “On the capacity of the ‘ether’ and cables in electrical communication,” *Proc. 1st All-Union Conf. Technological Reconstruction of the Commun. Sector and Low-Current Eng., Moscow*, 1933.
- [4] C. E. Shannon, “Communication in the presence of noise,” *Proceedings of the IRE*, vol. 37, no. 1, pp. 10–21, 1949.
- [5] B. Murmann, “ADC performance survey 1997–2010,” 2010, available online <http://www.stanford.edu/~murmman/adcsurvey.html>.
- [6] Y. Eldar and T. Michaeli, “Beyond bandlimited sampling,” *IEEE Signal Processing Magazine*, vol. 26, no. 3, pp. 48–68, Apr. 2009.
- [7] M. Unser, “Sampling – 50 years after Shannon,” *Proceedings of the IEEE*, vol. 88, no. 4, pp. 569–587, Apr. 2000.
- [8] E. Candes and T. Tao, “Near-optimal signal recovery from random projections: Universal encoding strategies?” *IEEE Transactions on Information Theory*, vol. 52, no. 12, pp. 5406–5425, Dec. 2006.
- [9] E. Candes, J. Romberg, and T. Tao, “Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information,” *IEEE Transactions on Information Theory*, vol. 52, no. 2, pp. 489–509, Feb. 2006.
- [10] D. Donoho, “Compressed sensing,” *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.
- [11] S. S. Chen, D. L. Donoho, and M. A. Saunders, “Atomic decomposition by basis pursuit,” *SIAM Review*, vol. 43, no. 1, pp. 129–159, 2001.
- [12] B. Hayes, “The best bits,” *American Scientist*, vol. 97, no. 4, pp. 276–280, Jul. 2009.

- [13] P. Marziliano, “Sampling innovations,” Ph.D. dissertation, Ecole Polytechnique Federale de Lausanne, Switzerland, 2001, no. 2369.
- [14] B. Murmann and B. Boser, “Digitally assisted analog integrated circuits,” *Queue*, vol. 2, pp. 64–71, Mar. 2004.
- [15] Wikipedia, “Ultra-wideband — Wikipedia, the free encyclopedia,” 2010, [Online; accessed 28-December-2010]. [Online]. Available: <http://en.wikipedia.org/w/index.php?title=Ultra-wideband&oldid=403061805>
- [16] B. Le, T. Rondeau, J. Reed, and C. Bostian, “Analog-to-digital converters,” *IEEE Signal Processing Magazine*, vol. 22, no. 6, pp. 69–77, Nov. 2005.
- [17] R. H. Walden, “Analog-to-digital converter survey and ana,” *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 4, pp. 539–550, Apr. 1999.
- [18] B. Murmann, “A/d converter trends: Power dissipation, scaling and digitally assisted architectures,” in *Custom Integrated Circuits Conference, 2008. CICC 2008. IEEE*, Sep. 2008, pp. 105–112.
- [19] P. Prandoni and M. Vetterli, “From Lagrange to Shannon ... and back: Another look at sampling,” *IEEE Signal Processing Magazine*, vol. 26, no. 5, pp. 138–144, Sep. 2009.
- [20] Y. C. Eldar, “Compressed sensing of analog signals in shift-invariant spaces,” *IEEE Transactions on Signal Processing*, vol. 57, pp. 2986–2997, Aug. 2009.
- [21] M. L. Yue and N. D. Minh, “Sampling signals from a union of subspaces,” *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 41–47, Mar. 2008.
- [22] E. Candes, “The restricted isometry property and its implications for compressed sensing,” *Comptes Rendus Mathematique*, vol. 346, no. 9-10, pp. 589–592, May 2008.
- [23] E. J. Candes and M. B. Wakin, “An introduction to compressive sampling,” *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 21–30, Mar. 2008.
- [24] E. Candes and J. Romberg, “Sparsity and incoherence in compressive sampling,” *Inverse Problems*, vol. 23, no. 3, p. 969, 2007.
- [25] R. G. Baraniuk, M. A. Davenport, R. A. DeVore, and M. B. Wakin, “A simple proof of the restricted isometry property for random matrices,” *Constructive Approximation*, vol. 28, no. 3, pp. 253–263, Dec. 2008.
- [26] E. Candes and J. K. Romberg, “ ℓ_1 -magic: Recovery of sparse signals via convex programming,” Caltech, Tech. Rep., 2005.

- [27] R. Chartrand and W. Yin, "Iteratively reweighted algorithms for compressive sensing," in *33rd International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2008.
- [28] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 1.21," <http://cvxr.com/cvx>, Sep. 2010.
- [29] M. F. Duarte, M. A. Davenport, D. Takhar, J. N. Laska, T. Sun, K. F. Kelly, and R. G. Baraniuk, "Single-pixel imaging via compressive sampling," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 83–91, Mar. 2008.
- [30] P. L. Dragotti, M. Vetterli, and T. Blu, "Sampling moments and reconstructing signals of finite rate of innovation: Shannon meets Strang-Fix," *IEEE Transactions on Signal Processing*, vol. 55, no. 5, Part 1, pp. 1741–1757, 2007.
- [31] M. Vetterli, P. Marziliano, and T. Blu, "Sampling signals with finite rate of innovation," *IEEE Transactions on Signal Processing*, vol. 50, no. 6, pp. 1417–1428, 2002.
- [32] S. Kirolos, J. Laska, M. Wakin, M. Duarte, D. Baron, T. Ragheb, Y. Massoud, and R. Baraniuk, "Analog-to-information conversion via random demodulation," in *IEEE Dallas Circuits and Systems Workshop (DCASW)*, Oct. 2006, pp. 71–74.
- [33] J. Laska, S. Kirolos, M. Duarte, T. Ragheb, R. Baraniuk, and Y. Massoud, "Theory and implementation of an analog-to-information converter using random demodulation," in *IEEE International Symposium on Circuits and Systems*, May 2007, pp. 1959–1962.
- [34] J. A. Tropp, J. N. Laska, M. F. Duarte, J. K. Romberg, and R. G. Baraniuk, "Beyond Nyquist: Efficient sampling of sparse bandlimited signals," *IEEE Transactions on Information Theory*, vol. 56, no. 1, pp. 520–544, Jan. 2010.
- [35] M. Mishali, Y. C. Eldar, and A. Elron, "Xampling—Part I: Practice," 2009, submitted to IEEE.
- [36] A. Björck, *Numerical Methods for Least Squares Problems*, 1st ed. Society for Industrial and Applied Mathematics, Dec. 1996.
- [37] M. F. Duarte, "Spectral compressive sensing toolbox," <http://dsp.rice.edu/scs>, 2010.
- [38] D. Needell and J. A. Tropp, "CoSaMP: Iterative signal recovery from incomplete and inaccurate samples," California Institute of Technology, Pasadena, Tech. Rep., 2008.

- [39] M. A. Davenport, P. T. Boufounos, M. B. Wakin, and R. G. Baraniuk, "Signal processing with compressive measurements," *Journal of Selected Topics in Signal Processing*, vol. 4, no. 2, pp. 445–460, Apr. 2010.
- [40] M. A. Davenport, J. N. Laska, P. T. Boufounos, and R. G. Baraniuk, "A simple proof that random matrices are democratic," Department of Electrical and Computer Engineering, Rice University, Tech. Rep., Nov. 2009.
- [41] J. N. Laska, P. T. Boufounos, M. A. Davenport, and R. G. Baraniuk, "Democracy in action: Quantization, saturation, and compressive sensing," Jul. 2009.
- [42] T. Ragheb, J. N. Laska, R. G. Baraniuk, and Y. Massoud, "A prototype hardware for random demodulation based compressive analog-to-digital conversion," in *51st Midwest Symposium on Circuits and Systems*, Knoxville, Aug. 2008, pp. 37–40.
- [43] M. F. Duarte and R. G. Baraniuk, "Spectral compressive sensing," Sep. 2010, submitted revised version to IEEE Transactions on Signal Processing.
- [44] R. R. Baraniuk, V. Cevher, M. F. Duarte, and C. Hedge, "Model-based compressive sensing," Department of Electrical and Computer Engineering, Rice University, Tech. Rep., 2009.
- [45] M. F. Duarte, C. Hegde, V. Cevher, and R. G. Baraniuk, "Recovery of compressible signals in unions of subspaces," in *Conference on Information Sciences and Systems (CISS)*, 2009.
- [46] P. Stoica and R. L. Moses, *Introduction to Spectral Analysis*. Prentice Hall, Feb. 1997.
- [47] T. Blu, P. Dragotti, M. Vetterli, P. Marziliano, and L. Coulot, "Sparse sampling of signal innovations," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 31–40, 2008.
- [48] J. Kusuma, A. Ridolfi, and M. Vetterli, "Sampling of communications systems with bandwidth expansion," in *IEEE International Conference on Communications*, vol. 3, 2002, pp. 1601–1605.
- [49] J. Kusuma, I. Maravic, and M. Vetterli, "Sampling with finite rate of innovation: Channel and timing estimation for UWB and GPS," in *IEEE International Conference on Communications*, vol. 5, May 2003, pp. 3540–3544.
- [50] I. Maravic, "Sampling methods for parametric non-bandlimited signals," Ph.D. dissertation, Ecole Polytechnique Federale de Lausanne, Switzerland, Lausanne, 2004.

- [51] I. Maravic and M. Vetterli, “Sampling and reconstruction of signals with finite rate of innovation in the presence of noise,” *IEEE Transactions on Signal Processing*, vol. 53, no. 8, pp. 2788–2805, 2005.
- [52] M. Vetterli, P. Prandoni, I. Maravic, and P. Marziliano, “FRI toolbox v1.0,” http://lcavwww.epfl.ch/research/topics/sampling_FRI.html, 2005.
- [53] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. D. Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen, *LAPACK Users’ Guide*, 3rd ed. Philadelphia, PA: Society for Industrial and Applied Mathematics, 1999.
- [54] J. Cadzow, “Signal enhancement—a composite property mapping algorithm,” *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 36, no. 1, pp. 49–62, Jan. 1988.
- [55] I. Schoenberg, “Cardinal interpolation and spline functions,” *J. Approx. Theory volume 2*, pp. 167–206, 1969.
- [56] G. Strang and G. Fix, “A Fourier analysis of the finite element variational method,” in *Constructive Aspects of Functional Analysis*, E. Cremonese, Ed., Rome, 1973, pp. 795–840.
- [57] M. Unser, “Splines: A perfect fit for signal/image processing,” *IEEE Signal Processing Magazine*, vol. 16, pp. 22–38, 1999.
- [58] M. Unser, A. Aldroubi, and M. Eden, “Polynomial spline signal approximations: Filter design and asymptotic equivalence with Shannon’s sampling theorem,” *IEEE Transactions on Information Theory*, vol. 38, no. 1, pp. 95–103, 1992.
- [59] M. Unser, “Cardinal exponential splines: Part II – think analog, act digital,” *IEEE Transactions on Signal Processing*, vol. 53, no. 4, pp. 1439–1449, 2005.
- [60] M. Unser and T. Blu, “Cardinal exponential splines: Part I – theory and filtering algorithms,” *IEEE Transactions on Signal Processing*, vol. 53, no. 4, pp. 1425–1438, 2005.
- [61] R. Tur, Y. C. Eldar, and Z. Friedman, “Low rate sampling of pulse streams with application to ultrasound imaging,” *CoRR*, vol. abs/1003.2822, 2010.
- [62] K. Gedalyahu, R. Tur, and Y. C. Eldar, “Multichannel sampling of pulse streams at the rate of innovation,” *CoRR*, vol. abs/1004.5070, 2010.
- [63] M. Mishali and Y. C. Eldar, “From theory to practice: Sub-Nyquist sampling of sparse wideband analog signals,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, no. 2, pp. 375–391, Apr. 2009.

- [64] —, “Blind multiband signal reconstruction: Compressed sensing for analog signals,” *IEEE Transactions on Signal Processing*, vol. 57, no. 3, pp. 993–1009, Mar. 2009.
- [65] M. Mishali and Y. Eldar, “Reduce and boost: Recovering arbitrary sets of jointly sparse vectors,” *IEEE Transactions on Signal Processing*, vol. 56, no. 10, pp. 4692–4702, Oct. 2008.
- [66] J. Chen and X. Huo, “Theoretical results on sparse representations of multiple-measurement vectors,” *IEEE Transactions on Signal Processing*, vol. 54, no. 12, pp. 4634–4643, Dec. 2006.
- [67] S. Cotter, B. Rao, K. Engan, and K. Kreutz-Delgado, “Sparse solutions to linear inverse problems with multiple measurement vectors,” *IEEE Transactions on Signal Processing*, vol. 53, no. 7, pp. 2477–2488, Jul. 2005.
- [68] Y. C. Eldar and M. Mishali, “Robust recovery of signals from a structured union of subspaces,” *IEEE Transactions on Information Theory*, vol. 55, no. 11, pp. 5302–5316, 2009.
- [69] J. A. Tropp, “Algorithms for simultaneous sparse approximation. part II: Convex relaxation,” *Signal Processing*, vol. 86, no. 3, pp. 589–602, 2006, sparse Approximations in Signal and Image Processing.
- [70] J. A. Tropp, A. C. Gilbert, and M. J. Strauss, “Algorithms for simultaneous sparse approximation: Part I: Greedy pursuit,” *Signal Processing*, vol. 86, pp. 572–588, Mar. 2006.
- [71] M. Mishali, Y. C. Eldar, O. Dounaevsky, and E. Shoshan, “Xampling: Analog to digital at sub-Nyquist rates,” Department of Electrical Engineering, Israel Institute of Technology, Tech. Rep., Dec. 2009.
- [72] Y.-P. Lin and P. P. Vaidyanathan, “Periodically nonuniform sampling of bandpass signals,” *IEEE Transactions on Circuits and Systems*, vol. 45, no. 3, pp. 340–351, Mar. 1998.
- [73] H. J. Landau, “Sampling, data transmission, and the nyquist rate,” *Proceedings of the IEEE*, vol. 55, no. 10, pp. 1701–1706, Oct. 1967.
- [74] M. Mishali and Y. Chen, “Software for the modulated wideband converter,” <http://www.technion.ac.il/~moshiko/software.html>, Nov. 2010.
- [75] Y. Chen, M. Mishali, Y. C. Eldar, and A. O. Hero, “Modulated wideband converter with non-ideal lowpass filters,” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. IEEE, Mar. 2010, pp. 3630–3633.

- [76] M. Mishali, A. Elron, and Y. C. Eldar, “Sub-Nyquist processing with the modulated wideband converter,” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. IEEE, Mar. 2010, pp. 3626–3629.
- [77] T. Dickson, E. Laskin, I. Khalid, R. Beerkens, J. Xie, B. Karajica, and S. Voinescu, “An 80-Gb/s $2^{31} - 1$ pseudorandom binary sequence generator in SiGe BiCMOS technology,” *IEEE Journal of Solid-State Circuits*, vol. 40, no. 12, pp. 2735–2745, Dec. 2005.
- [78] M. Mishali and Y. C. Eldar, “Wideband spectrum sensing at sub-Nyquist rates,” Department of Electrical Engineering, Israel Institute of Technology, Tech. Rep., 2010.
- [79] T. Blu, P.-L. Dragotti, P. Vetterli, M. and Marziliano, and L. Coulot, “Sparse sampling of signal innovations,” *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 31–40, Mar. 2008.
- [80] S. Kirolos, T. Ragheb, J. Laska, M. Duarte, Y. Massoud, and R. Baraniuk, “Practical issues in implementing analog-to-information converters,” in *The 6th International Workshop on System-on-Chip for Real-Time Applications*, Dec. 2006, p. 141.
- [81] G. Acosta and D. M. A. Ingram, “OFDM simulation using MATLAB,” Smart Antenna Research Laboratory, Georgia Institute of Technology, Tech. Rep., Aug. 2000.
- [82] J. Wepman, “Analog-to-digital converters and their applications in radio receivers,” *IEEE Communications Magazine*, vol. 33, no. 5, pp. 39–45, May 1995.
- [83] K. Gedalyahu and Y. C. Eldar, “Time-delay estimation from low-rate samples: A union of subspaces approach,” *IEEE Transactions on Signal Processing*, vol. 58, no. 6, pp. 3017–3031, 2010.
- [84] P. Dyke, *An Introduction to Laplace Transforms and Fourier Series*. Springer, Nov. 1999.
- [85] S. Mallat, *A Wavelet Tour of Signal Processing*. London: AP Professional, 1997.
- [86] M. J. C. S. Reis, P. J. S. G. Ferreira, and S. F. S. P. Soares, “Linear combinations of B-splines as generating functions for signal approximation,” *Digital Signal Processing*, vol. 15, no. 3, pp. 226–236, May 2005.
- [87] L. Ahlfors, *Complex Analysis*, 3rd ed. McGraw-Hill, 1 1979, no. Science/Engineering/Math, science/Engineering/Math.