ECOLE POLYTECHNIQUE FEDERALE DE LAUSANNE

Finite Rate of Innovation sampling technics for embedded UWB devices.

Yann Barbotin

advised by

Pr. Martin VETTERLI (EPFL/LCAV),

Dr. Amal EKBAL (QUALCOMM/CORP.R&D)

A thesis submitted in partial fulfillment for the
degree of Master of Science

in the

SCHOOL OF COMPUTERS AND COMMUNICATIONS SYSTEMS (IC)
Department of Communication Systems

March 2009

Declaration of Authorship

I, *Yann Barbotin*, declare that this thesis titled, "*Finite Rate of Innovation sampling technics for embedded UWB devices*" and the work presented in it are my own. I confirm that:

- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Date: *Friday, March 13 2009*

ECOLE POLYTECHNIQUE FEDERALE DE LAUSANNE

Abstract

SCHOOL OF COMPUTERS AND COMMUNICATIONS SYSTEMS (IC)
Department of Communication Systems MSc

Yann Barbotin

This report studies the applicability of Finite Rate of Innovation (FRI) algorithms to UltraWide Band (UWB) communications, more precisely in the scope of Low Power Body Area Networks (LP-BAN). Three main issues are studied and given proposed solutions.

First, the classical FRI algorithm is modified to accomodate different symmetrical pulse shapes. Such a modification – necessary to get acceptable performances – is done by a simple equalization. Second, LP-BAN devices limitations such as drift, jitter and aggressive quantization are blended in the algorithm. It is done by adjusting the equalization template and development of a suited quantization algorithm. Third and last, the cost of FRI denoising procedure (Cadzow denoising) is greatly reduced to fit the requirements of a low power embedded device. It is centered on performing most of the computations in a low-dimension Krylov subspace of the matrix to be denoised. The particular structure of the projected matrix enables selective computation of the eigenpairs.

The result is an algorithm able to resolve close paths within a reasonnable computational budget. Some issues remain on quantization.

KEYWORDS: **Finite Rate of Innovation, FRI, UltraWide Band, UWB, Low-Power, Body Area Network, ranging, equalization, quantization, Krylov subspaces**

Acknowledgements

This has been a motivating project, and these six months at Qualcomm have been wonderful. On the Qualcomm side, thanks go to my manager Amal, my mentor Jun, David the project leader and fellow "ranging-mate" Cristian who provided the LP-BAN signal generator. The way the whole team works on a challenging project, while maintaining a friendly atmosphere is very nice.

On the academical side, I want to thanks Pr. Vetterli and Pr. Blu for the great amount of help and attention provided to get the project going, especially during the "hard" time at the beginning. Your patience was much appreciated!

And last, to my parents and my brother and sister Pierre-Yves and Solène, for being such a great family. I am blessed! The Christmas break at home reloaded the motivation'o'meter necessary to write this leeeenngthy report.

Contents

List of Tables

Abbreviations

UWB	**U**ltra**W**ide **B**and ($> 500MHz$)
LP-BAN	**L**ow **P**ower **B**ody **A**rea **N**etwork
FRI	**F**inite **R**ate (of) **I**nnovation
Rx	The **R**eceiving device
LPF	**L**ow-**P**ass **F**ilter
BPF	**B**and-**P**ass **F**ilter
AWGN	**A**dditive **W**hite **G**aussian **N**oise
RMSE	**R**oot **M**ean **S**quare **E**rror
DFT	**D**iscrete **F**ourier **T**ransform
FFT	**F**ast **F**ourier **T**ransform
SVD	**S**ingular **V**alue **D**ecomposition
PSD	**P**ower **S**pectral **D**ensity
SNR	**S**ignal (to) **N**oise **R**atio
NSR	**N**oise (to) **S**ignal **R**atio
DoF	**D**egree(s) **o**f **F**reedom
s/**ns**	**s**tationnary/**n**on-**s**tationnary
CR bound	**C**ramér-**R**ao (lower) bound
MC quantization	**M**onte**C**arlo quantization
MT quantization	**M**ultiple **T**hresholds quantization
ppm	**p**arts **p**er **m**illion
LS	**L**east **S**quares
TLS	**T**otal **L**east **S**quares
sTLS	**s**tructured **T**otal **L**east **S**quares
PRO	**P**artial **R**e**O**rthogonalization
LAPACK	**L**inear **A**lgebra **PACK**age

Symbols

$[\cdot]$, $(\cdot)$	discrete and continuous indexing
a, b, c, $\ldots$, α, β, γ, $\ldots$	a scalar
i, j, k, l, m, n, p	often used for indexing purpose
A, B, C, $\ldots$	a (usually) constant scalar
$\boldsymbol{a}$, $\boldsymbol{b}$, $\boldsymbol{c}$, $\ldots$, $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, $\boldsymbol{\gamma}$, $\ldots$	a vector
$\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$, $\ldots$, $\boldsymbol{\Gamma}$, $\boldsymbol{\Lambda}$, $\ldots$	a matrix
$i : j$	a range, similar to MATLAB notation
$\cdot^*$	the Hermitian conjugate
$\mathrm{Sp}\{\cdot\}$	the trace operator (from the german "Spur")
$\perp$	orthogonality (relation on vectors or polynoms
	or property for a matrix or a sequence of polynoms)
$\boldsymbol{\Xi}$, $\boldsymbol{\Lambda}$; $\boldsymbol{\xi}$, λ	eigenvectors, eigenvalues matrix; an eigenvector, an eigenvalue
θ	usually a parameter in the context of estimation
	or a Ritz value in Rayleigh-Ritz theory context
$\hat{\cdot}$	Fourier transform in signal processing context,
	or an estimator in estimation context
$\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$, $\ldots$	a space
$\mathrm{proj}_{\mathcal{S}}\cdot$	projection of matrix or a vector "$\cdot$" in space $\mathcal{S}$
ω, $e^{j\omega}$	pulsation and periodic pulsation in a Fourier analysis context
$\mathbb{E}[\cdot]$	the expectation (borelian operator)
$\mathrm{Var}[\cdot]$	the variance (borelian operator)
$\Re$, $\Im$	real and imaginary part

Chapter 0

Introduction

0.1 The ranging problem in Ultra Wide Band (UWB) communications

UWB communications are radio communications using a frequency bandwidth larger than 500MHz. In comparison to narrow-band communications which rely on modulation of a carrier frequency, the large bandwidth of UWB communications allows to send signals with features well-localized in time – the more localized is a signal in time the more it spreads in frequency. This opens the door to communications based on pulses, and information can be encoded in the distance between pulses (Pulse Position Modulation: PPM) or in their amplitude (PAM) or the pulse width (PWM). One of the key advantage of pulse based communication is the ability to precisely localize the time of arrival of the information (the pulse). An interesting application is to measure the distance between two UWB devices, and it is called *ranging*. Take an example between devices A and B, a potential 2-way protocol may be:

- A and B agree they will do the ranging procedure
- A sends a pulse to B and keep a timestamp t_0 of sending time
- B receives the pulse and estimates finely the time t_1 at which it received it
- B sends a pulse back to A recording the sending time t_2
- A receives the pulse and estimates finely the time t_4 at which the pulse was received
- B transmits $\Delta t = t_2 - t_1$ to A
- A estimates the time of flight between B and himself as $ToF_{A,B} = \frac{t_4 - t_0 - \Delta t}{2}$ and multiply by the propagation speed to estimate its distance to B

The accuracy of the ranging relies on a good estimation of t_1 and t_4, *i.e.* a good estimation of the pulse location in time. In practice the channel on which the pulse propagates will produce echos, and so the problem becomes an estimation of the first received pulse location, which we assume has followed a straight path between A and B. It is called the *Line of Sight* (LoS) pulse. Figure 1 shows a channel with two strong echos and a weak LoS. In case a scenario as described is figure 1 is relevant, one needs to retrieve precisely

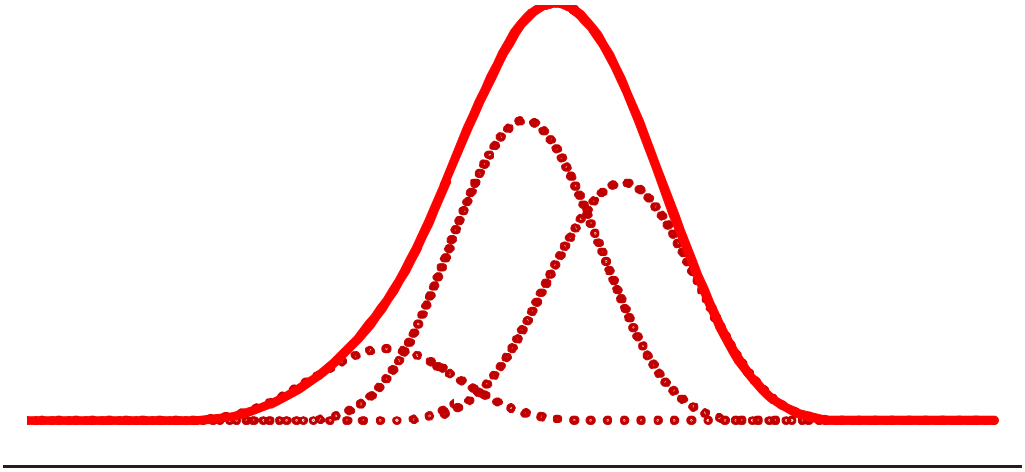


FIGURE 1: A channel with two echos.

the location of the overlaping pulses, and any algorithm based on maximum detection is doomed to fail. We turn our attention to a parametric method called *Finite Rate of Innovation* (FRI). [1–3] In order for a parametric method to be succesful, one shall be able to have a relatively accurate model of the pulse shape. We will see it may not be the case for cost and power consumption reasons. The question "is a parametric method relevant?" shall be kept in mind at any time, doing so we will identify up to which point FRI may be succesfully applied.

0.2 Short introduction to FRI

The goal of this section is not be a comprehensive review of the theory as numerous references exist. [1–3] We proceed to quickly restate the basics. The intuition is that a signal with a finite number of unknown parameters – the degrees of freedom (DoF) – may be fully characterized by as many samples providing an adequate sampling kernel. Such an observation is not constructive, and the hardest part is to formulate an algorithm to retrieve the parameters from the sample.

In the case of a periodic pulse train, several algorithms were formulated respectively for the sinc kernel [1–3], the gaussian kernel [2], B-splines [2, 4], E-splines [4], and the list goes on. A major requirement is for the algorithm to be robust to noise provided extra

samples. Robust algorithms are available in the sinc and E-spline cases [1, 4]. Emphasis is put on the sinc sampling kernel as it is the canonical one for FRI and it is relevant to the UWB setup.

0.2.1 FRI with a sinc sampling kernel

This section is taken from [1].

Noiseless case A periodic train of pulses is defined as:

$$x(t) = \sum_{k=1}^{K} \sum_{l \in \mathbb{Z}} c_k \delta(t - t_k - l\tau). \tag{1}$$

,such that K is the number of pulses per period τ and $c_k,\ t_k$ their amplitude and location. Observing $x(t)$ through a sampling device operating at frequency $1/T$ and with a sinc sampling kernel of bandwidth B yields the samples:

$$y_n = \langle x(t), \mathrm{sinc}(B(nT - t)) \rangle = \sum_{k=1}^{K} x_k D_B(nT - t_k) \ , \quad n = 1 \dots N. \tag{2}$$

D_B is the Dirichlet kernel of bandwidth B: $D_B(t) = \frac{\sin(\pi B t)}{B\tau \sin(\pi t/\tau)}$. In a few words, periodicity has been transfered from the signal to the sampling kernel. N is taken odd for simplicity.

The Fourier coefficients of $x(t)$ (x is periodic) falling within the bandwidth of the Dirichlet kernel can be retrieved from the Fourier coefficients of y_n:

$$\hat{y}_m = \sum_{n=1}^{N} y_n e^{-j2\pi mn/N} = \begin{cases} \tau \hat{x}_m & \text{if } |m| \leq \lfloor B\tau/2 \rfloor \\ 0 & \text{else} \end{cases} \tag{3}$$

These coefficients verify:

$$\hat{x}_m = \frac{1}{\tau} \sum_{k=1}^{K} x_k e^{-j2\pi m t_k/\tau}. \tag{4}$$

Each coefficient of the Fourier series depends on a very small subset of the Fourier vectors (the complex exponentials). If one knows the subset, which means knows the location, finding the amplitude is a simple change of basis, *i.e.* finding the solution of a linear system.

The hard part is to find the set of complex exponentials, which is a non-linear problem. It is done by the *Annihilating filter* method. The annihilating filter is in the z-domain:

$H(z) = \sum_{k=0}^{K} h_k z^{-k} = \prod_{k=1}^{K} \left(1 - e^{-j2\pi t_k/\tau} z^{-1}\right)$. It has the property to annihilate the spectrum of x:

$$h_m * \hat{x}_m = 0. \tag{5}$$

The coefficients $\{h_i\}_{i=1\dots K},\ h_0 = 1$ are solution of the toeplitz system:

$$\begin{pmatrix} \hat{x}_{-1} & \dots & \hat{x}_{-K} \\ \vdots & \ddots & \vdots \\ \hat{x}_{K-2} & \dots & \hat{x}_{-1} \end{pmatrix} \begin{pmatrix} h_1 \\ \vdots \\ h_K \end{pmatrix} = - \begin{pmatrix} \hat{x}_0 \\ \vdots \\ \hat{x}_{K-1} \end{pmatrix}. \tag{6}$$

Once the filter coefficients are obtained, the locations are computed finding its roots.

Noisy case: Total Least Squares & Cadzow denoising In case the samples are corrupted, additional samples are required to denoise the signal prior to the annihilating filter computation. In the Cadzow denoising procedure, the DFT coefficients of the samples are arranged in a Toeplitz matrix A. We restrain ourselves to a square matrix, even if it is not strictly necessary:

$$\boldsymbol{A} = \begin{pmatrix} \hat{y}_0 & \dots & \hat{y}_{-(N+1)/2} \\ \vdots & \ddots & \vdots \\ \hat{y}_{(N+1)/2} & \dots & \hat{y}_0 \end{pmatrix}. \tag{7}$$

Note that for a real input signal, $\boldsymbol{A}$ is also hermitian. If the original signal contains K distinct pulses, $\boldsymbol{A}$ shall be of rank K. This can be enforced by clipping the $(N+1)/2-K$ smallest eigenvalues of $\boldsymbol{A}$ to 0 and synthetizing $\tilde{\boldsymbol{A}}$ from this "partial" eigenvalue decomposition. $\tilde{\boldsymbol{A}}$ has the property to be the closest rank K matrix to $\boldsymbol{A}$ in the Frobenius norm. However, $\tilde{\boldsymbol{A}}$ is not Toeplitz anymore. It is made toeplitz by averaging the diagonals. The process of reducing to rank K and "toeplitzation" is repeated until the $K+1^{th}$ eigenvalue gets significantly smaller than the K^{th} one. For a complete study of the convergence of this algorithm and thorough argumentation, see [5].

Building a topelitz matrix $\boldsymbol{T}$ as in equation 6 with column dimension $K+1$, and annihilating filter $\boldsymbol{h}$ of degree $K+1$ yields the homogeneous equation: $\boldsymbol{Th} = \boldsymbol{0}$, which is to say $\boldsymbol{h}$ belongs to the null-space of $\boldsymbol{T}$. In case $\boldsymbol{T}$ is "tall", its null-space may be empty. With or without prior denoising, it is reasonnable to solve a relaxed annihilating equation. The TLS solution of such a surdetermined system is $\boldsymbol{h}_{TLS} = \arg\min_{\|\boldsymbol{h}\|^2=1} \|\boldsymbol{Th}\|^2$. The solution may be found in the null-space of the closest approximation of $\boldsymbol{T}$ (in the Frobenius norm) with a non empty null-space. Such a vector is colinear to the last column of $\boldsymbol{V}$ assuming $\boldsymbol{T} = \boldsymbol{USV}^T$.

Chapter 1

FRI in a non-ideal setup

1.1 Problem statement & overview

FRI theory exposed in chapter 0 gives the theoretical foundations to build upon. It provides a toolbox to work with rather than an immediate "blackbox" solution. Indeed, real-life impulse communications cannot afford the infinite support of the *sinc* function, and implementation has its constraints dictated by available technology or cost.

This chapter provides a case-study of FRI implementation for an LP-BAN (Low Power Body Area Network) platform. It is not tractable to start with an exact model of LP-BAN as too many modifications from the theory would be introduced at once. The progressive approach applied summarizes in two points:

- effects on FRI performances of constraints inherent to UWB communications (square demodulation, pulse shape, ...), and possible solutions.
- effects on FRI performances of hardware limitations (drift, jitter, quantization, ...), and possible solutions.

The typical Rx hardware chain is illustrated in figure 1.1. Capital letters label different parts of the chain, and as a convenient reminder, D falls just after discretization in *time*. We use D_Q rather than E after quantization (discretization in *amplitude*). With this reference map, we are ready to proceed with the first part of the chapter.

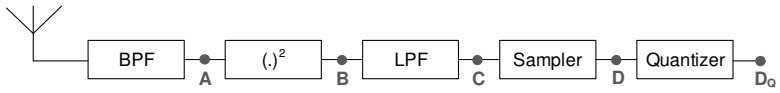


FIGURE 1.1: Simplified diagram of the receiver (Rx) radio-frequency frontend.

1.2 FRI in the general UWB setup

1.2.1 The Rx chain model

1.2.1.1 Signal analysis

Consider the periodic analog signal:

$$\begin{aligned} y(t) &= x(t) + \epsilon(t) \ , \\ &\text{such that} \qquad \forall t \in [0\tau] : \ x(t) \in \mathbb{R}, \ \ \epsilon(t) = \epsilon_{\Re}(t) + j\epsilon_{\Im}(t) \qquad (1.1) \\ &\qquad\qquad \epsilon_{\Re}, \epsilon_{\Im} \text{ iid white gaussian wss process : } \ \epsilon_{\Re}, \epsilon_{\Im} \sim \mathcal{N}(0, \sigma^2). \end{aligned}$$

Just before demodulation, the noise ϵ_A has lost is whiteness:

$$\begin{aligned} y_A(t) &= BPF\{y(\cdot)\}(t) && (1.2) \\ &\overset{def}{=} x_A(t) + \epsilon_A(t) \ , && (1.3) \\ \text{PSD}\left\{\epsilon_A(\cdot_1)\right\}(e^{j\omega}) &\overset{wss}{=} \mathcal{F}\left\{\mathbb{E}\left[\epsilon_A(\cdot_2)\epsilon_A^*(\cdot_2 - \cdot_1)\right]\right\}(e^{j\omega}) && (1.4) \\ &= |h_{BPF}(e^{j\omega})|^2 \mathcal{F}\left\{\mathbb{E}\left[\epsilon(\cdot_2)\epsilon^*(\cdot_2 - \cdot_1)\right]\right\}(e^{j\omega}) && (1.5) \\ &= |h_{BPF}(e^{j\omega})|^2. && (1.6) \end{aligned}$$

Then at point C – before discretization – the signal is made of 3 principal components:

$$\begin{aligned} y_C(t) = & \quad LPF\{x_A(\cdot_)x_A^*(\cdot_)\}(t) \\ & + \ 2 \cdot LPF\{x_A(\cdot_)\Re\left[\epsilon_A(\cdot_)\right]\}(t) \qquad (1.7) \\ & + \ LPF\{\epsilon_A(\cdot_)\epsilon_A^*(\cdot_)\}(t), \\ \overset{def}{=} & \quad x_C(t) + \epsilon_{\text{ns}}(t) + \epsilon_{\text{s}}(t). \end{aligned}$$

The noise has a stationary part ϵ_{s} and a non-stationnary one ϵ_{ns}. This distinction is important as it will call for two different mathematical treatments.

1.2.1.2 Modelisation of the stationnary noise $\epsilon_{\mathbf{s}}$

It is well-known the sum of squares of iid and normally distributed random variables follows a χ^2 distribution. However for a general quadratic form, the distribution is a mixture of χ^2 distributions [6, 7]. To show it, we work in the discrete domain using matrix formalism. The discretized original noise process is written $\boldsymbol{\epsilon}$, and to each filter $filter\_name$ is associated the convolution mask $\boldsymbol{c}_{filter\_name}$. Then:

$$\begin{aligned} \epsilon_B &= (\boldsymbol{c}_{BPF}\boldsymbol{\epsilon})^* \boldsymbol{c}_{BPF}\boldsymbol{\epsilon} && (1.8)\\ &= \boldsymbol{\epsilon}^* \underbrace{\boldsymbol{c}^*_{BPF}\boldsymbol{c}_{BPF}}_{\text{call it } \boldsymbol{P}} \boldsymbol{\epsilon}. && (1.9) \end{aligned}$$

Assuming $\boldsymbol{P}$ is non-singular, it is a positive-definite quadratic form of normally distributed random variables. It is shown in [7, 8] that a positive-definite quadratic form of 0-mean gaussian random variables with non-singular covariance structure follows a mixture of χ^2. Indeed, a vector of correlated gaussian random variables can be seen as the transformation of an iid vector of $\mathcal{N}(0,1)$ by $\sqrt{\boldsymbol{\Sigma}}$, $\boldsymbol{\Sigma}$ the covariance matrix of the original vector: $\boldsymbol{x}_{corr} = \sqrt{\boldsymbol{\Sigma}}\boldsymbol{x}_{iid}$. Since covariance matrices are symmetric, the quadratic form in equation 1.8 is equivalent to:

$$\epsilon_B = \boldsymbol{\epsilon}^*_{idd}\sqrt{\boldsymbol{\Sigma}}\boldsymbol{P}\sqrt{\boldsymbol{\Sigma}}\boldsymbol{\epsilon}_{\boldsymbol{iid}}. \quad (1.10)$$

In our case, $\boldsymbol{\epsilon}$ is made of iid random variables, so its covariance matrix is the identity and equation 1.10 is unecessarily complicated. However it is mentionned to highlight the iid property of the noise is not a requirement. Then using a the diagonalization of $\boldsymbol{P}$ by an orthonormal matrix $\boldsymbol{\Xi}$, $\boldsymbol{P} = \boldsymbol{\Xi}^*\boldsymbol{\Lambda}\boldsymbol{\Xi}$, each stationnary noise sample verifies:

$$\begin{aligned} \epsilon_B &= \boldsymbol{\epsilon}^*\boldsymbol{P}\boldsymbol{\epsilon} \\ &= \boldsymbol{Q}\boldsymbol{\epsilon}^*\boldsymbol{\Lambda} \underbrace{\boldsymbol{Q}\boldsymbol{\epsilon}}_{\substack{\boldsymbol{Q} \text{ is } \perp \Rightarrow \text{iid}, \\ \sim\mathcal{N}(0,1)}} && (1.11)\\ &= \sum_{i=1}^{n} \lambda_i \psi_i \ , \quad \psi_i \sim \chi^2_1. \end{aligned}$$

which proves the mixture of χ^2 property. Call c_k the k^{th} moment, $c_k \overset{def}{=} \mathrm{Sp}\left\{\boldsymbol{P}^k\right\} = \sum \lambda_i^k l_i$, such that l_i is the number of degrees of freedom (DoF) of each random variable. It is proven in [9] ϵ_B is well approximated by a χ^2 distributed random variable.

Proposition 1.1. ***[Liu et al. 2009]*** *Let be an approximation of a mixture of χ^2 by a χ^2 distribution with l DoF having the same skewness . This approximation differs in kurtosis as:*

$$\Delta_\kappa = 12 \left| \frac{1}{l} - \frac{c_4}{c_2^2} \right| .$$

Proof. See [9]. □

Of course, in case $\lambda_1 = \lambda_2 = \cdots = \lambda_n$ one gets a perfect fit as $c_2^2/c_4 = n = l$. In general the fitness is dictated by the homogeneity of $\boldsymbol{P}$'s eigenvalues. They can be derived from the equation of thelpbandpass filter.

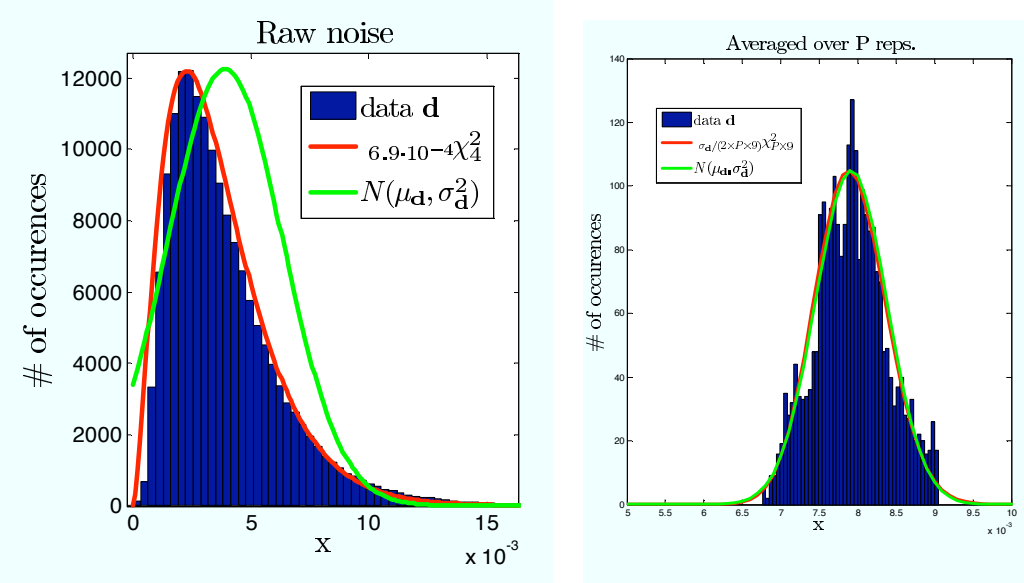
Similarly, assuming the approximation holds for ϵ_B and the filter $LPF\{\cdot\}$ has non-negative coefficients $[\dots f_i \dots]$, filtering after demodulation by the square filter and the low-pass filter yields another mixture of χ^2 distributed random variables, with a circulant autocorrelation matrix. Since the filter coefficients are assumed non-negative, the same approximation may be applied. No precise claim is made on the validity of the approximation as it depends on the filters specification. However, simulations show a χ^2 fit is relevant – see figure 1.2(a).

Usually, several sequences $\boldsymbol{x}_D^{(1)}, \ \dots, \ \boldsymbol{x}_D^{(P)}$ are obtained at different but relatively close times: different to have independence between the $\boldsymbol{x}_D^{(\cdot)}$ but close enough to have propagated on the same channel thus having identical distribution. Adding them coherently results in a sum of (non weighted) χ^2 distributed random variables, which has the property to increase the number of DoF. As the number of DoF increase, the χ^2 distribution quickly tends to a gaussian one. This fact can more generally be considered true from the central limit theorem. Such a property is witnessed in 1.2(b) for a value of P typical to LP-BAN .

All the preceeding developments lead to the approximation:

$$\text{For } P \text{ large enough, } \sum_{p=1}^{P} \epsilon_D^{(p)} \text{ is normally distributed.} \tag{1.12}$$

To fully characterize the process, the autocorrelation is computed emprically. Figure 1.3 shows *Pearson moment-product* for the stationnary noise (normalized autocorrelation)



(a) Stationary noise in LP-BAN at the end of the Rx chain. (b) Averaging pulses increase the number of DoF in the χ^2 distribution, allowing for a gaussian approximation.

FIGURE 1.2: Stationary noise distribution.

with or without averaging of the pulses:

$$\rho_{\mathrm{s}}[k,i] = \frac{\mathbb{E}\left[\epsilon_{\mathrm{s}}[k]^*\epsilon_{\mathrm{s}}[k-i]\right]}{\sqrt{\mathrm{Var}[\epsilon_{\mathrm{s}}[k]]\mathrm{Var}[\epsilon_{\mathrm{s}}[k-i]]}} \tag{1.13}$$

$$= \frac{\mathbb{E}\left[\epsilon_{\mathrm{s}}[0]^*\epsilon_{\mathrm{s}}[-i]\right]}{\mathrm{Var}[\epsilon_{\mathrm{s}}]} \tag{1.14}$$

$$= \frac{r_{\mathrm{s}}[i]}{r_{\mathrm{s}}[0]}. \tag{1.15}$$

ρ_{s} only depends on filters in the Rx chain. It can thus be estimated over a long noise sequence of fixed power. Then autocorrelation for a particular noise of power $r_{\mathrm{s}}[0]$ – estimated with the standard unbiased estimator of the variance – is computed as $r_{\mathrm{s}}[i] = r_{\mathrm{s}}[0]\rho_{\mathrm{s}}[i]$. It overcomes the large variance of the autocorrelation estimator for large indices.

We thus have a complete characterization of the stationnary noise component by estimating its mean and variance.

1.2.1.3 Modelisation of the non-stationnary noise ϵ_{ns}

The non-stationnary component has the good taste to be gaussian. Given an original noise power σ^2, a convolution matrix $\boldsymbol{C}_{filter\_name}$ and a gain g_{filter_name} for each filter (possible presence of an active electronic component), the autocorrelation matrix $\boldsymbol{R}_{\mathrm{ns}}$ of $\boldsymbol{\epsilon}_{\mathrm{ns}}$ verifies:

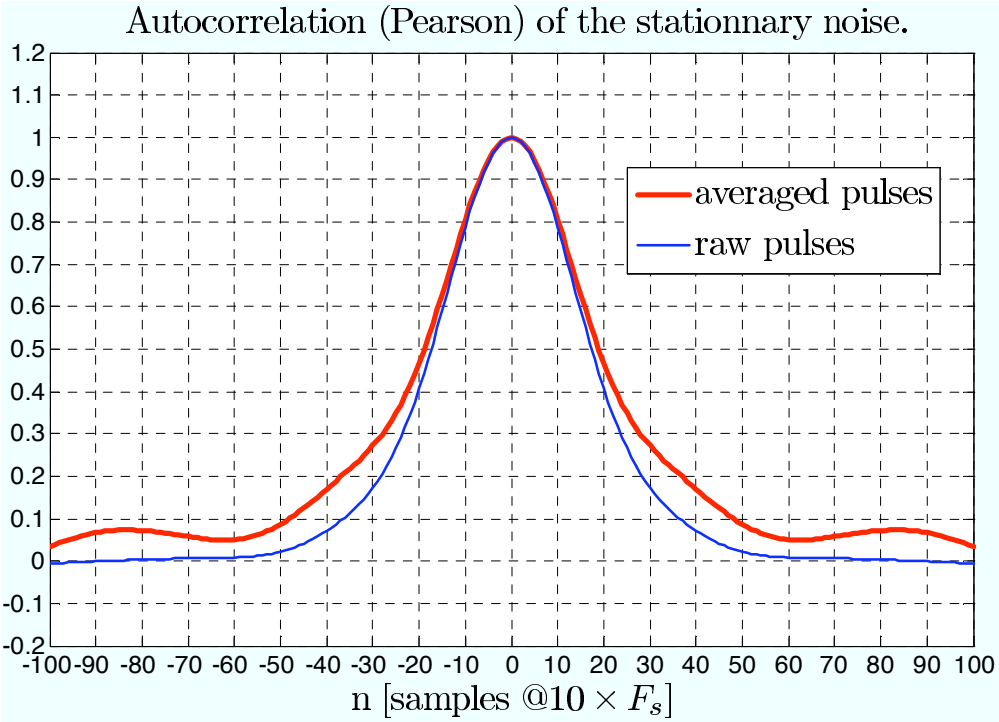


FIGURE 1.3: Pearson moment-product of the stationnary noise. F_s is LP-BAN Rx sampling frequency.

$$\boldsymbol{R}_{\text{ns}} = 4\sigma^2 g_{BPF}^4 g_{LPF}^2 \boldsymbol{C}_{LPF} \left[(\boldsymbol{C}_{BPF} \boldsymbol{x}\boldsymbol{x}^* \boldsymbol{C}_{BPF}^*) \cdot (\boldsymbol{C}_{BPF} \boldsymbol{C}_{BPF}^*) \right] \boldsymbol{C}_{LPF}^*. \tag{1.16}$$

where $\cdot$ is the *Hadamard product*, *a.k.a.* "element-wise product" or "direct product".

The non-stationnarity of the process is caused by "$\boldsymbol{x}\boldsymbol{x}^*$" not being toeplitz. The relevance of a circular convolution is questionable, however the signal is well localized, a shift can make sure the non-0 elements are not wrapped around. The primary effect of averaging is to multiply the autocorrelation by $\frac{1}{P}$ – the proof is trivial.

1.2.1.4 Validation of the noise model

In the rest of the chapter we will assume P pulses are acquired and averaged. Pulses could be kept distinct for a variant of FRI described in [10], however averaging of the pulses is necessary in LP-BAN to virtually increase the number of quantization bits as later explained in 1.3.1.

The noise random processes ϵ_{s} and ϵ_{ns} are obviously cross-correlated. Simulations showed a Pearson moment-product not exceeding 0.15. We thus make the assumption these 2 processes are not cross-correlated: $\boldsymbol{R}_{\text{s,ns}} = \boldsymbol{0}$; still keeping in mind the crudeness of

the approximation. Hence it will be meaningless to give more than a couple significant digits in the autocorrelation of $\boldsymbol{\epsilon}_D$.

The global autocorrelation matrix is:

$$\begin{aligned} \boldsymbol{R} &= \mathbb{E}\left[\boldsymbol{\epsilon}_D \boldsymbol{\epsilon}_D^*\right] && (1.17)\\ &= \begin{pmatrix} \mathbb{I} & \mathbb{I} \end{pmatrix} \begin{pmatrix} \boldsymbol{R}_{\mathrm{s}} & \boldsymbol{R}_{\mathrm{s,ns}} \\ \boldsymbol{R}_{\mathrm{ns,s}} & \boldsymbol{R}_{\mathrm{ns}} \end{pmatrix} \begin{pmatrix} \mathbb{I} \\ \mathbb{I} \end{pmatrix} && (1.18)\\ &= \boldsymbol{R}_{\mathrm{s}} + \cancelto{0}{\boldsymbol{R}_{\mathrm{s,ns}}} + \cancelto{0}{\boldsymbol{R}_{\mathrm{ns,s}}} + \boldsymbol{R}_{\mathrm{ns}}. && (1.19) \end{aligned}$$

To illustrate the above formula, figure 1.4 shows $\boldsymbol{R}$ structure for an input signal with one pulse. A distinctive bulge on the diagonal is observed at the pulse position.

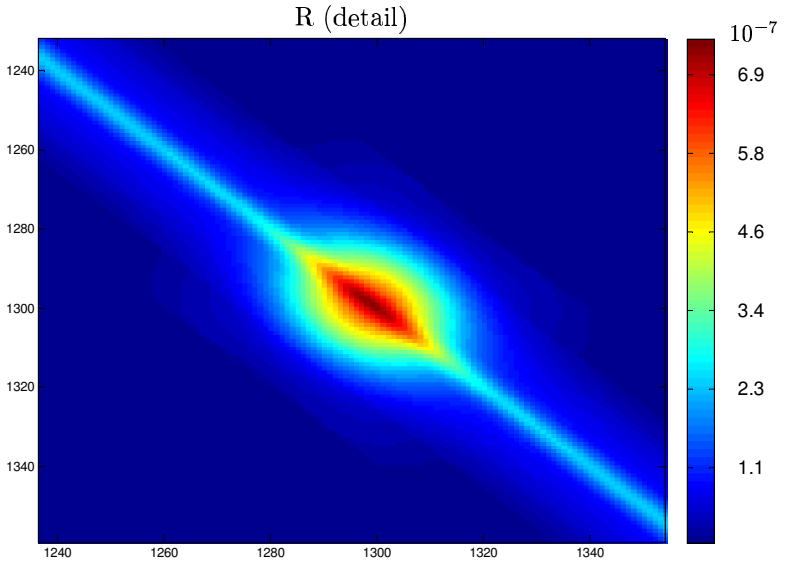


FIGURE 1.4: The autocorrelation matrix for a single pulse input. Frequency is $10 \times F_s$.

The adequacy of the noise model is witnessed in figure 1.5. The component 1.5.d was obtained by substraction of the signal and the stationnary noise from the noisy signal, $(d) = (a) + (b) - (c)$. They seem all compatible with the model.

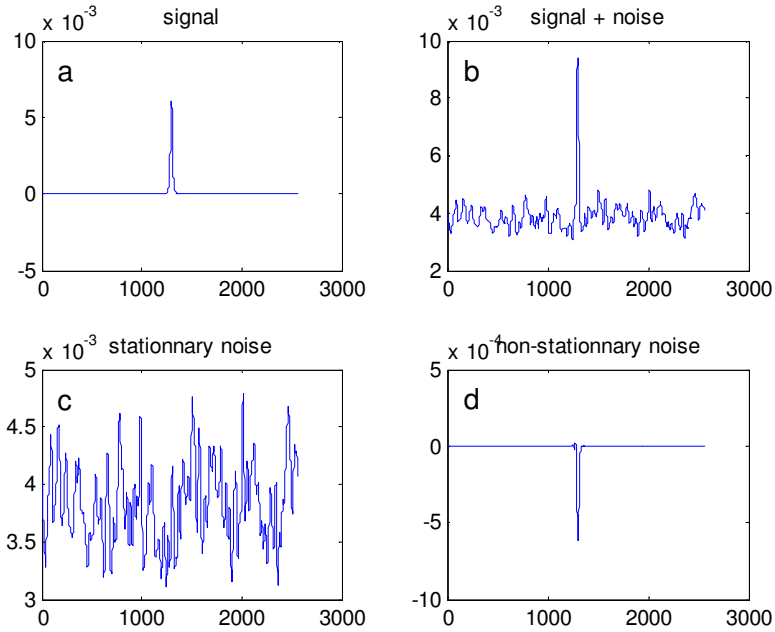


FIGURE 1.5: Components of the signal.

1.2.1.5 The LP-BAN pulse shape

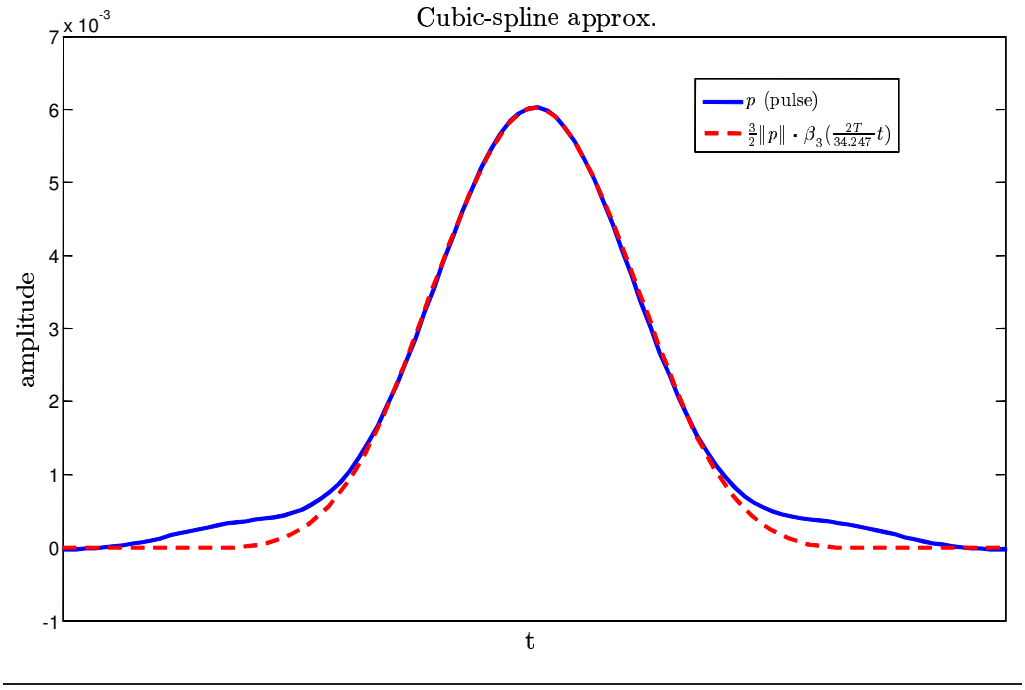


FIGURE 1.6: Pulse fit by a cubic B-spline .

The last component of the signal is the pulse in itself. The studied pulse is $x_C(t)$ in the analog domain, and $x_D[n]$ in the digital domain. It is well approximated by a cubic B-spline .(figure 1.6) B-spline are naturally defined in the Fourier domain (Schönberg formula [11]):

$$\hat{\beta}^n(\omega) = \left(\frac{\sin(\omega/2)}{\omega/2}\right)^{n+1} = \left(\frac{e^{j\omega/2} - e^{j\omega/2}}{j\omega}\right)^{n+1}. \tag{1.20}$$

As in [12], introducing the one-sided power function $(x)_+^n \stackrel{def}{=} x^n$ if $x \geq 0$, $= 0$ else and its Fourier transform $X_+^n(\omega)$, the identity $\frac{X_+^n(\omega)}{n!} = \frac{1}{(j\omega)^{n+1}}$ yields the time-domain formula [12]:

$$\beta^n(x) = \frac{1}{n!}\sum_{k=0}^{n+1}\binom{n+1}{k}(-1)^k\left(x - k + \frac{n+1}{2}\right)_+^n. \tag{1.21}$$

where the monom of complex exponentials was expanded by the binomial formula.

By equation 1.21, the pulse approximation is in the time-domain (up to scaling and dilation):

$$\beta^3(x) = \begin{cases} \frac{2}{3} - |t^2| + \frac{|t^3|}{2} & , \quad 0 \leq |t| < 1 \\ \frac{(2-|t|)^3}{6} & , \quad 1 \leq |t| < 2 \\ 0 & , \quad 2 \leq |t| \end{cases}. \tag{1.22}$$

1.2.2 Computation of the Cramér-Rao (CR) bound

1.2.2.1 Generalities (theory)

With the knowledge of a good approximation of the noise process, its autocorrelation and the pulse shape, the next step is to compute a theoretical lower bound on the variance of the paramters estimates. For unbiased estimator this bound is called the *Crámer-Rao bound*. A measure for the amount of information a random vector Y carries about a given parameter θ is called *Fisher information* $\mathcal{I}_Y(\theta)$:

$$\mathcal{I}_Y(\theta) = \mathbb{E}\left[\frac{\partial^2}{\partial\theta^2}\ln f_\theta(Y)\right]. \tag{1.23}$$

such that $f_\theta(Y)$ is the joint probability density function of Y, parametrized by θ.

Then any unbiased estimator $\hat{\theta}_Y$ of θ based on a realization of Y has its variance bounded by: ([13] §32.8)

$$\mathrm{Var}\left[\hat{\theta}_Y\right] \geq \left[\mathcal{I}_Y(\theta)\right]^{-1}. \tag{1.24}$$

The above inequality is the one stated by Cramér in his seminal book. [13] It corresponds to the univariate case or multivariate with independant estimation of the parameters. For independant *and* joint estimation in the multivariate setup:

Theorem 1.2. ***[Cramér-Rao lower bound]***
Given a random vector Y, the vector of samples, with a joint probability density $f_{\boldsymbol{\theta}} \in C^1$ and a set of parameters $\boldsymbol{\theta} = [\theta_1 \dots \theta_n]$. Define its Fisher information matrix $\boldsymbol{J}_Y(\boldsymbol{\theta})$ such that $J_{k,l} = \mathbb{E}\left[\frac{\partial}{\partial\theta_k}\ln f_{\boldsymbol{\theta}}(Y)\frac{\partial}{\partial\theta_l}\ln f_{\boldsymbol{\theta}}(Y)\right]$. The covariance matrix of any unbiased estimator $\hat{\boldsymbol{\theta}}_Y$ of $\boldsymbol{\theta}$ based on Y is bounded by:

$$Cov\left\{\hat{\boldsymbol{\theta}}_Y\right\} \geq \boldsymbol{J}_Y(\boldsymbol{\theta})^{-1}. \tag{1.25}$$

Proof. See Rao's original paper [14]. □

An explicit formula for the Fisher information matrix of a signal with additive gaussian noise is found in [15] and [1]:

Proposition 1.3. *Given a timeseries $\boldsymbol{y}_{\boldsymbol{\theta}} = \boldsymbol{x}_{\boldsymbol{\theta}} + \boldsymbol{\epsilon}_{\boldsymbol{\theta}}$, with $\boldsymbol{x}_{\boldsymbol{\theta}}$ deterministic and $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}$ a gaussian 0-mean random vector with autocorrelation matrix $\boldsymbol{R}_{\boldsymbol{\theta}}$, the entries of the Fisher information matrix $\boldsymbol{J}_Y(\boldsymbol{\theta})$ are:*

$$J_{k,l} = \frac{1}{2}Sp\left\{\boldsymbol{R}_{\boldsymbol{\theta}}^{-1}\frac{\partial\boldsymbol{R}_{\boldsymbol{\theta}}}{\partial\theta_k}\boldsymbol{R}_{\boldsymbol{\theta}}^{-1}\frac{\partial\boldsymbol{R}_{\boldsymbol{\theta}}}{\partial\theta_l}\right\} + \left(\frac{\partial\boldsymbol{x}_{\boldsymbol{\theta}}}{\partial\theta_k}\right)^* \boldsymbol{R}_{\boldsymbol{\theta}}^{-1}\frac{\partial\boldsymbol{x}_{\boldsymbol{\theta}}}{\partial\theta_l}. \tag{1.26}$$

Proof. See [15]. □

Note that in general the "trace" term in equation 1.26 vanishes as the noise is independant of the estimated parameters. This is however not the case in the LP-BAN setup as the non-stationnary noise is correlated with the input signal and thus the parameters. The intuition is the spike-like nature of the noise at the location of the pulse provides additional information.

1.2.2.2 Computation of the CR bound on pulse locations in LP-BAN signals

CR formula for LP-BAN From equation 1.26, it looks we have all the ingredients to compute a CR bound on the pulse locations in LP-BAN signals. The only missing link is the derivative of the pulse shape. Recalling the pulse is a cubic B-spline :

$$\begin{aligned}\frac{\partial\beta^3}{\partial t}(t) &= \mathcal{F}^{-1}\left\{\mathcal{F}\left\{\frac{\partial\beta^3}{\partial\cdot_2}(\cdot_2)\right\}(\cdot_1)\right\}(t)\\ &= \mathcal{F}^{-1}\left\{\hat{\beta}^2(\cdot_)\left(e^{j\cdot_/2}-e^{-j\cdot_/2}\right)\right\}(t)\\ &= \left(\beta^2 * \Delta_c^1\right)(t)\\ &= \beta^2\left(t+\frac{1}{2}\right)-\beta^2\left(t-\frac{1}{2}\right).\end{aligned} \tag{1.27}$$

where $\Delta_c^1 = \delta\left(\frac{1}{2}\right) - \delta\left(-\frac{1}{2}\right)$ is the centered finite difference operator By formula 1.21, the quadratic B-spline is in the time-domain:

$$\beta^2(x) = \begin{cases} -|t|^2+\frac{3}{4} & , \quad 0 \le |t| < \frac{1}{2} \\ \frac{(|t|-\frac{3}{2})^2}{2} & , \quad \frac{1}{2} \le |t| < \frac{3}{2} \\ 0 & , \quad \frac{3}{2} \le |t| \end{cases}. \tag{1.28}$$

If we make the approximation the signal is a sum of pulses $\phi(t) = a\beta^3(s\cdot t)$ and noise as in [1], the signal has the form:

$$y(t) = x_{\boldsymbol{t}_K}(t) + \text{noise} = \sum_{k=1}^{K} c_k\phi(t-t_k) + \text{noise}. \tag{1.29}$$

Then we identify the different terms in equation 1.26 to compute the Fisher information matrix of the pulse locations $\boldsymbol{t}_K = [t_k]_{k=1:K}$ based on samples obtained at time $\boldsymbol{n}/F_s = [1\dots\ N]^T/F_s$.

- $\frac{\partial \boldsymbol{x}_{\boldsymbol{\theta}}}{\partial\theta_k} = [-c_k\phi'(1/F_s - t_k)\cdots - c_k\phi'(N/F_s - t_k)]^T$,

$$\text{s.t. } \phi'(t) = a \cdot s \cdot \left[\beta^2\left(s \cdot t + \tfrac{1}{2}\right) - \beta^2\left(s \cdot t - \tfrac{1}{2}\right)\right].$$

- $\frac{\partial \boldsymbol{R}}{\partial t_k} = \frac{\partial \boldsymbol{R}_{\text{ns}}}{\partial t_k}$
$$= 4\sigma^2 g_{BPF}^2 g_{LPF}^2 \boldsymbol{C}_{LPF}\left[\boldsymbol{B}_k \cdot (\boldsymbol{C}_{BPF}\boldsymbol{C}_{BPF}^*)\right]\boldsymbol{C}_{LPF}^*,$$
$$\text{s.t. } \boldsymbol{B}_k = -a_{BPF}^2 \cdot s_{BPF}^2\left(\tilde{\boldsymbol{B}}_k + \tilde{\boldsymbol{B}}_k^*\right),$$
$$\text{and } \tilde{\boldsymbol{B}}_k = \beta^{3\prime}\left(s_{BPF}[\boldsymbol{n}/F_s - t_k]\right)\beta^3\left(s_{BPF}[\boldsymbol{n}/F_s - t_k]\right)^*.$$

This is all we need to compute the CR bound.

Validation of the formula (and efficience of the FRI based algorithm) To validate the formula, we used a "toy example" signal made of a single pulse at location t_1. Then over 200 trials, we measured the RMSE of the FRI based estimation. recall several approximations were made through the bound computation. Thus, what we call CR bound is in fact an approximation of the real CR bound.

We consider the bound *valid* if the RMSE of the FRI algorithm is larger than the bound for all tested SNR. Note it is only a sanity check, it does not gives a definitive answer, just raise the confidence in the computed bound correctness.

We consider the FRI based algorithm to be *efficient* if it *kisses* the CR bound. Keep in mind it is only a toy example. Being efficient on this signal is a prerequisite to good performances on multi-tap signals.

Two approximations of the bound were used:

- A "good" approximation, faithful to the formula developped above. However since the autocorrelation matrix $\boldsymbol{R}_{\text{s}}$ is empirically estimated and truncated, it may not be exactly *positive definite* (*pd*) as seen in 1.7. The problem is $\boldsymbol{r}_1^*\boldsymbol{R}^{-1}\boldsymbol{r}_2$ is not an inner-product anymore, which may result in negative Fisher information, which does not make sense. It is quite natural to obtain a non-sensical result as we started with a non-pd autocorrelation matrix... The solution employed, was to clip negative eigenvalues to the smallest positive one. Over several tests, $\|\boldsymbol{R}_{\text{s}}\|_\infty$ varied by less than 3%.

- The second approximation called "crude" does not enforce positive definitness of the autocorrelation matrix. When negative Fisher informations occures, and thus results in an imaginary Cramer-Rao bound, we treat this value as a real number. the second deviation from orthodoxy is the omition of the trace term in the Fisher information formula. The "crude" approximation will thus be above the "good" one, and may show some variations. The purpose of this approximation is to

verify enforcing positive definitness has a neglectable impact, and to see if the trace-term is important to the bound accuracy.

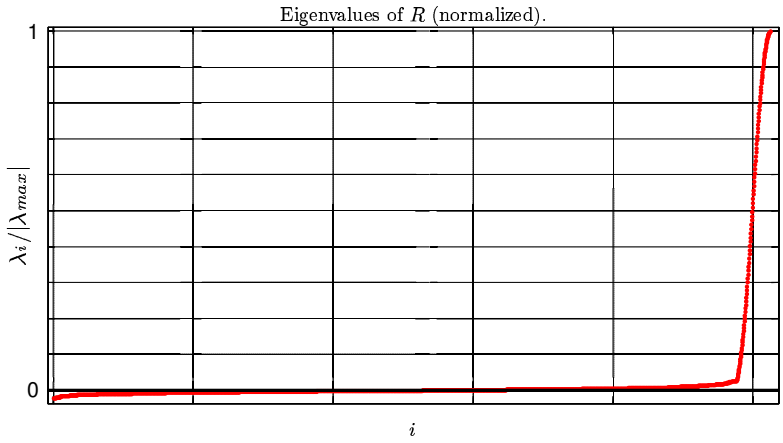
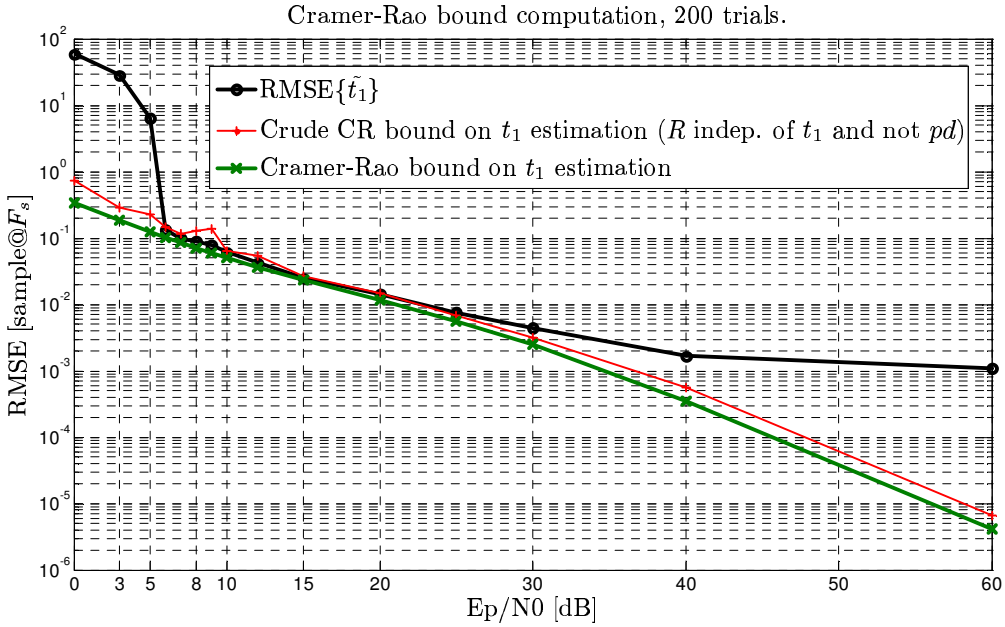


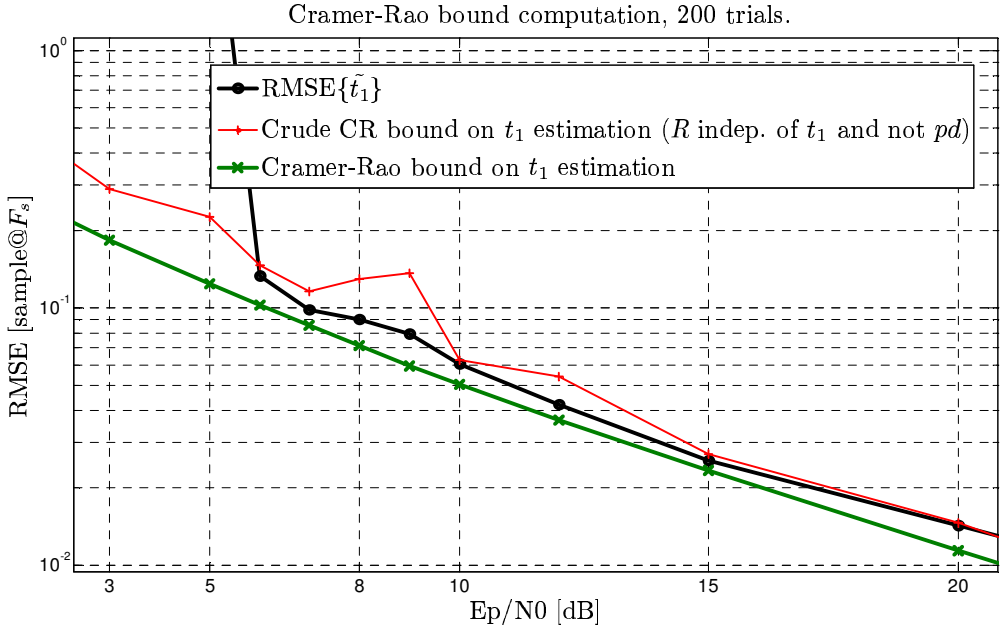
FIGURE 1.7: The estimated autocorrelation matrix may not be positive definite.

Results are reported in figures 1.8(a) and 1.8(b). It seems the trace-term has a small impact, but its lack is enough to fail the CR bound validation.

The "good" bound approximation passes validation and the FRI based algorithm is efficient down to $E_p/N0 = 6dB$. It was excpected the algorithm will diverge from the bound at high SNR as the pulse origin was determined by "naked-eyed" analysis on a pulse-shape sampled at $10 \times F_s$ by fitting a B-spline template on the pulse shape as in figure 1.6. Thus, the result is slightly biased.



(a) CR bound on the $Ep/N0$ interval from 0dB to 60dB



(b) Zoom on 1.8(a) in the $Ep/N0$ interval from 3dB to 20dB

FIGURE 1.8: Comparison of the FRI algorithm RMSE with the "good" and "crude" bounds.

1.2.3 The multi-tap channel: adding equalization to the FRI algorithm

1.2.3.1 Problem statement

The name *multi-tap channel* is used for LP-BAN signals as only one pulse is transmitted, the other ones observed being echos from the channel.

One of the first problem arising with multitap channels is due to the non-linearity of the Rx chain. Indeed, the demodulation block is non-linear. If we input a signal made of two real-valued pulses $s(t) = p_1(t) + p_2(t)$ the demodulation introduces a cross-term:

$$s_{\text{demod}}(t) = p_1^2(t) + p_2^2(t) + 2p_1(t)p_2(t). \tag{1.30}$$

This additional term is non null if the supports of p_1 and p_2 overlap. It is obviously an important setback, but it is ignored for now. In fact it will surface later in the results (figure 1.15), limiting the accuracy for close paths.

Another important issue is the mismatch between the real pulse shape and the pre-supposed sinc kernel. This mismatch did not prevented the FRI algorithm to kiss the Cramér-Rao bound in the single tap case since the TLS solutions coincide for different symmetrical pulses. However, as visible in figure 1.9(b), it is not the case for multitap anymore. Blame cannot be put on the cross-term as the pulses support barely overlap in figure 1.9(b). The problem is nevertheless easy to spot, the TLS solution is not the expected one but a symmetrical mixture of sinc pulses around the main pulse as seen in figure 1.9(c). In a few words, more energy is removed from the residual by cancelling the side lobes of the sinc pulses than by fitting a smaller pulse. It is a catastrophic outcome as small pulses are missed, and worse the large one as well. Moreover a simple misestimation of the number of pulses in the signal will lead to the same result. It seems essential to get somewhat closer to the ideal sinc shape.

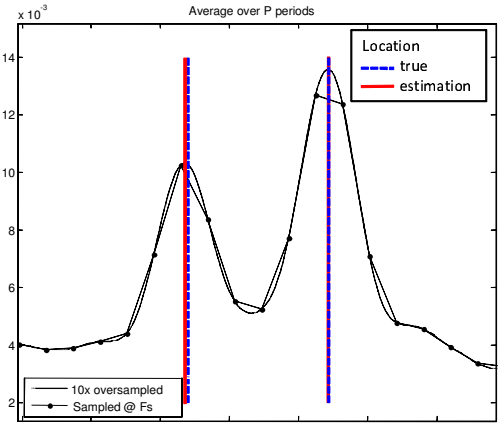
1.2.3.2 Equalization of the spectrum

Since the pulse p is symmetric, $\exists w$ symmetric s.t.:

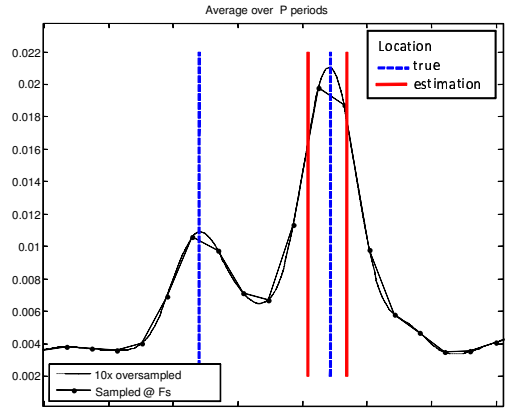
$$(p * w)[n] = sinc_B[n]. \tag{1.31}$$

with B the desired bandwidth. In the DFT domain:

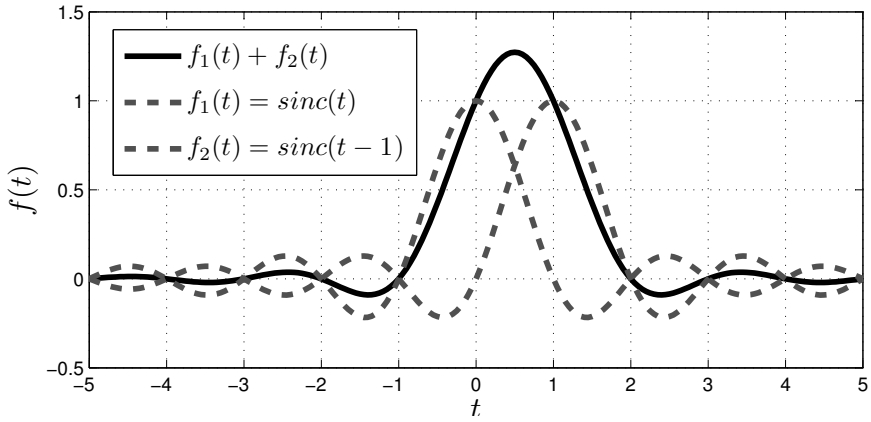
$$\begin{aligned} & P[\omega]W[\omega] &=& \text{rect}_B[\omega] \\ \Leftrightarrow \quad & W[\omega] &=& \tfrac{\text{rect}_B[\omega]}{P[\omega]}. \end{aligned} \tag{1.32}$$



(a) Plain FRI algorithm on a 2-taps signal



(b) Plain FRI algorithm on the same signal, with a smaller first tap



(c) Explanation of the phenomenon in figure 1.9(b).

FIGURE 1.9: Application of the plain FRI algorithm with an important model mismatch is not suitable for multitap signals.

If we choose the bandwidth B to be maximal – *i.e.* the largest bandwidth avoiding aliasing – equation 1.32 simplifies to:

$$W[\omega] = \frac{1}{P[\omega]}. \tag{1.33}$$

It is simply the inverse of the pulse spectrum. In the noisy case, it is customary to use the *Wiener filter* for its RMSE minimization properties. In the LP-BAN setup:

$$y_D[n] = x_D[n] + \epsilon_{\mathrm{s}}[n] + \epsilon_{\mathrm{ns}}[n]. \tag{1.34}$$

One cannot apply directly the Wiener filter as ϵ_{ns} is non-stationnary. Its PSD is thus singular at the origin, and its treatment would require regularization to annihilate the singularity. Choice is made not to wander down that path. Instead, the more pragmatic approach to ignore it, and pray for the best will be employed

Regarding the stationnary noise, the equalization has a whitening effect. Its autocorrelation is similar to the pulse shape as seen in figure 1.3. This makes the equalization relatively well-suited for the task as the sTLS solution is energy-wise optimal for samples corrupted by additive white gaussian noise. This equalization operation can be seen as a deconvolution. Figure 1.10[1] shows we observe the desired dirichlet kernel shaped pulse (periodic sinc) through a device with a B-spline shaped point-spread function. Application of a simple deconvolution – multiplication by the inverse in the Fourier domain – yields the original dirichlet pulse plus a whitened noise in the low-frequencies. With this simple aparatus half of the DFT coefficients can be used in the FRI algorithm.

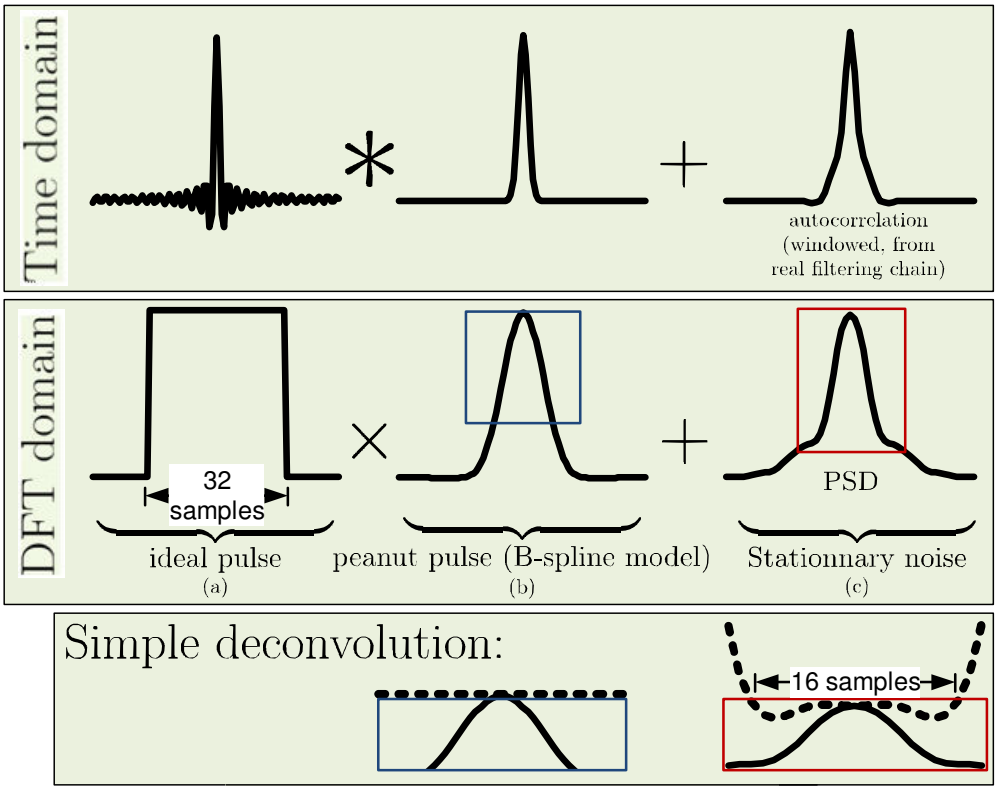


FIGURE 1.10: Straightforward deconvolution whitened the noise

Thus, for a maximal bandwidth and ignoring the non-stationnary noise, the Wiener deconvolution filter is:

$$W[\omega] = \frac{P^*[\omega]}{P[\omega]P^*[\omega] + \mathrm{NSR}[\omega]}. \tag{1.35}$$

Where $\mathrm{NSR}[\omega]$ is the spectral *Noise to Signal Ratio*. Assuming the original sinc pulse has full bandwidth, the convolution filter p has unit-norm and the DFT of the Pearson moment-product $S[\omega]$ is known – which is reasonable as it only depends on the Rx chain

[1]The signals shown in this figure are slightly oversampled to make them smoother. In reality, the dirichlet kernel would have maximum bandwidth, *i.e.* its DFT spectrum would be a flat line instead of a box

filters – the spectral NSR is:

$$\mathrm{NSR}[\omega] = \frac{\sigma^2}{\eta^2} S[\omega]. \tag{1.36}$$

, where σ^2 is the noise power and η^2 is the signal power. The noise power is evaluated on a noise-only portion of the input, and the signal power on portion containing some pulses after substraction of the noise floor.

1.2.3.3 Numerical results

While the Wiener deconvolution approach is certainly a good one, the simple deconvolution followed by discard of the high frequencies Fourier coefficients proved to be quite simple yet efficient in the simulations. As expected, equalization made the FRI algorithm to perform very well on multitap signals, figure 1.11 shows the benefits. Equalization imposes itself so clearly, it will be implied from now on.

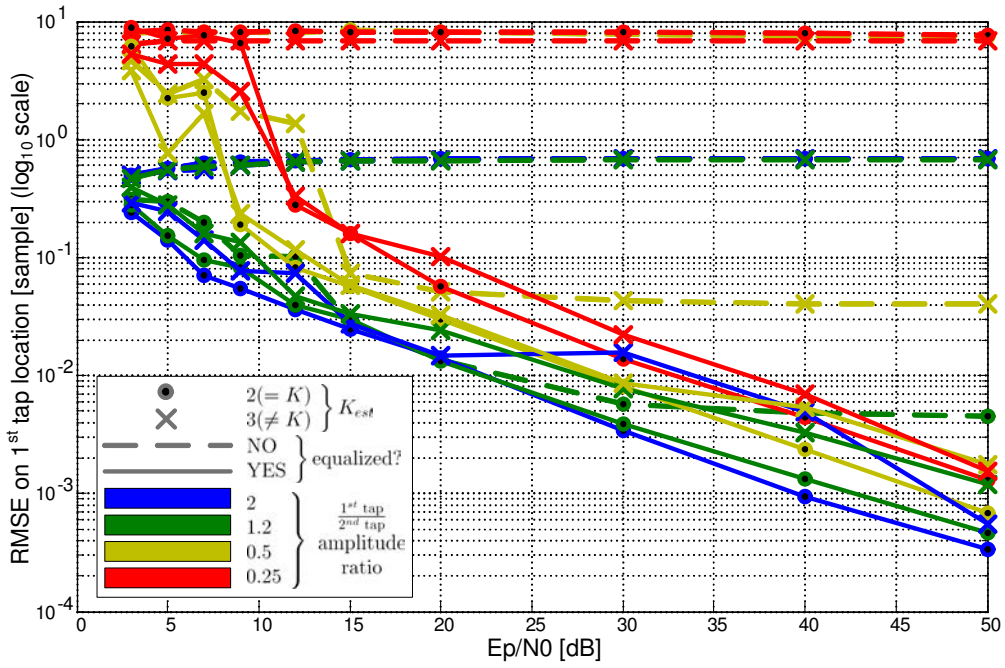


FIGURE 1.11: Equalization of the coefficients improves performances (2 distant taps, RMSE computed on 50 trials).

1.3 Low-power/low-cost UWB receiver: aggressive quantization, drift and jitter

1.3.1 FRI with 1-bit quantization

Full double precision was assumed in previous results. A more stringent quantization is a reality on embedded devices. There is no better way to find if the algorithm is "fire-proof" than by using the minimum number of bits possible: 1. The goal is to obtain more than 1-bit of information by repeating several time the same signal. Each repetion shall provide a new point of view, be it by a different realization of the noise or a different quantization level. Two approaches will be described: one based solely on the noise, and one hybrid.

The goal is to draw a rough pictures of these solutions, as time does not allow for more. The subsequent analysis has some rough edges as well.

1.3.1.1 Monte-Carlo (MC) quantization

Monte-Carlo sampling, is a sampling scheme where some randomness is included in the process. A famous example is the *Buffon's needle* experiment. It can be seen as MC sampling if its goal is to estimate π. MC sampling for LP-BAN is very similar, it consist of fixing a threshold τ and through P repetitions of the same signal – *i.e.* different noise realizations – collect for each sample a binary sequence $(b_1, \dots, b_P)$. Since repetition are independant, the only exploitable quantity in the binary sequence is the occurence rate of each digit – we call r the rate of "1" in the sequence, which defines the rate of "0" as $1 - r$. Then, given this rate, use a reconstruction function $f : [0\ 1] \mapsto \mathcal{S}$. A trivial example would be $f(r) = r$, in which case $\mathcal{S} = [0\ 1]$.

The maximum likelihood estimator of the rate based on $(b_1, \dots, b_P)$ realisation of the i.i.d $(B_1, \dots, b_P)$ is $\hat{r}_P = \frac{1}{P}\sum_{i=1}^{P} b_i$. It bias depends on r, however $\max_r \left(\mathbb{E}\left[\hat{r}_P - r\right]\right) \geq \frac{1}{2(P+1)}$. It is then consistent since the bias vanishes with P. Its variance is $\mathrm{Var}\left[\hat{r}\right] = \frac{\sigma^2}{P}$, with $\sigma^2 = \mathrm{Var}\left[B_i\right]$. Let $\theta \in [0\ \theta_{\max}]$ an input value to be quantized with the P-repetition MC quantizer. Call $\hat{\theta}_P$ the output of this estimator.

For a particular value θ, every repetition –*i.e.* every non-quantized samples spaced by the pulse reptition period – is the sum of:

- the noiseless signal value θ
- a χ^2 random variable independant of θ

- a 0-mean gaussian random variable with variance roughly $\theta\sigma_d$ (roughly because it depends on adjacent samples value as well).

For the sake of simplicity, the χ^2 r.v. will be assumed gaussian with mean μ_χ and variance σ_χ^2. This simplification thus only allows to give qualitative results. The noise is then aggregated in a single gaussian r.v. with mean $\mu = \mu_\chi$ and variance $\sigma^2 = \theta\sigma_d + \sigma_\chi^2$. Given a single fixed threshold τ, the binary random variables $B_i \sim \text{Bernoulli}\left(Q_{\mu,\sigma^2}(\tau - \theta)\right)$. It follows the variance of the estimator $\hat{r}$ is:

$$\begin{aligned} \text{Var}\left[\hat{r}\right] &= \frac{Q_{\mu,\sigma^2}(\tau-\theta)\left(1 - Q_{\mu,\sigma^2}(\tau-\theta)\right)}{P}. && (1.37) \\ &= \frac{r(1-r)}{P} && (1.38) \end{aligned}$$

It is not insightful to base the variance of an estimator on a particular value of the estimated parameter. We recall $\theta \in [0\ \theta_{\max}]$ – it is noteworthy $\theta_{\max}$ is proportional to the SNR. A Taylor expansion followed by first order approximation on the variance of $\hat{\theta}$ yields:

$$\begin{aligned} \text{Var}\left[\hat{\theta}\right] &\overset{\text{Taylor}}{\approx} \left(Q^{-1\prime}_{\mu,\sigma^2}\left(\mathbb{E}\left[\hat{r}\right]\right)\right)^2 \text{Var}\left[\hat{r}\right] \\ &\overset{2}{=} \left(Q'^{-1}_{\mu,\sigma^2}\left(Q^{-1}_{\mu,\sigma^2}\left(\mathbb{E}\left[\hat{r}\right]\right)\right)\right)^2 \text{Var}\left[\hat{r}\right] \\ &= \frac{Q_{\mu,\sigma^2}(\tau-\theta)\left(1 - Q_{\mu,\sigma^2}(\tau-\theta)\right)}{P\mathcal{N}_{\mu,\sigma^2}\left(Q^{-1}_{\mu,\sigma^2}(r)\right)} \\ &= \frac{Q_{\mu,\sigma^2}(\tau-\theta)\left(1 - Q_{\mu,\sigma^2}(\tau-\theta)\right)}{P\mathcal{N}_{\mu,\sigma^2}(\tau-\theta)}. \end{aligned} \quad (1.39)$$

, with $\mathcal{N}$ the gaussian pdf. For SNR$\to \infty$, *i.e.* $\theta_{\max}$ large, the quantity $max_\theta\,|\tau - \theta - \mu| \to \theta_{\max} \propto$ SNR. The conclusion is:

Proposition 1.4. *As the SNR grows, the variance of the monte-carlo estimator in the range of interest grows like*

$$|\theta| \propto SNR: \quad Var\left[\hat{\theta}_P\right] \sim \frac{e^{|\theta|}}{\sqrt{|\theta|}P}.$$

Proof. $\forall t < \mu : 1 < \frac{Q_{\mu,\sigma^2}(t)}{\mathcal{N}_{\mu,\sigma^2}(t)} < 3$. The easy way to show it is to use the integral criterion for monotonic positive functions and to bound the equivalent series using the identity

[2] for a continuous, invertible and differentiable function g and its inverse g^{-1}, $g^{-1\prime}(t) = \frac{1}{g'\left(g^{-1}(t)\right)}$. It is a simple and direct consequence of the chain rule for differentiation.

$1 + 1/2 + 1/4 + \cdots = 2$. Thus for $|\theta|$ large enough assuming $-\theta \approx \tau - \theta - \mu < 0$ (the same holds for positve, the roles of Q and $1 - Q$ are swaped) , equation 1.39 gives:

$$\frac{1 - Q_{\mu,\sigma^2}(\tau - \theta)}{P\mathcal{N}_{\mu,\sigma^2}(\tau - \theta)} < \text{Var}\left[\hat{\theta}_P\right] < 3\frac{1 - Q_{\mu,\sigma^2}(\tau - \theta)}{P\mathcal{N}_{\mu,\sigma^2}(\tau - \theta)}$$

which means:

$$\text{Var}\left[\hat{\theta}_P\right] \sim \frac{1}{P\mathcal{N}_{0,\sigma^2}(\theta)}$$

Then recall $\sigma^2 \propto |\theta|$. □

It is thus impossible to use this kind of sampling for any input SNR, as its variance grows exponentially – the bias of $\hat{r}_P$ not even taken into account, the RMSE will be even larger than the variance.

1.3.1.2 Multiple Thresholds (MT) quantization

Using the Monte-carlo sampling at high SNR is bound to produce catastrophic results. This section has the purpose to present an algorithm well-behaved at high SNR. A good target is uniform quantization for noiseless signal. It can be achieved using a different threshold for each repetition.

To do so, the maximum amplitude of the signal must be determined in as few repetitions as possible. A possible way to do it is:

- Start with a low initial threshold τ_1 and multiply it by 2 for each repetition if some digit is "1" – $\tau_{\text{new}} = 2\tau_{\text{current}}$ – otherwise repeat quantization with the same threshold a few times.
- If the quantized signal is all-0 for these repetitions, conclude current threshold is too high and take $\tau_{\text{new}} = (\tau_{\text{current}} - \tau_{\text{previous}})/2$, otherwise conclude current threshold is too low, $\tau_{\text{new}} = 2\tau_{\text{current}}$.
- Stop when the uncertainty on the signal amplitude is smaller than the achievable unfiorm quantization step, *i.e.* $\tau_{\text{current}} - \tau_{\text{previous}} \geq \frac{\tau_{\text{current}}}{\#\text{ of repetitions remaining}}$, call current threshold value $\tau_{\max}$.
- Use the remaining repetions to do uniform quantization in $[\tau_1\ \tau_{\max}]$.

At high SNR, the result is equivalent to uniform quantization, up to a few repetions "wasted" to estimate the signal amplitude. In order to perform correctly in a medium range of SNR, each level of the "uniform quantization" can be repeated. In any case,

the output of this algorithm is a vector of thresholds $\boldsymbol{\tau}$ and their associated number of repetitions $\boldsymbol{p}$, and a matrice $\boldsymbol{R}$ where each row represents a rate of "1" for a particular threshold associated. The task is then to devise a function f_{optimal} such that

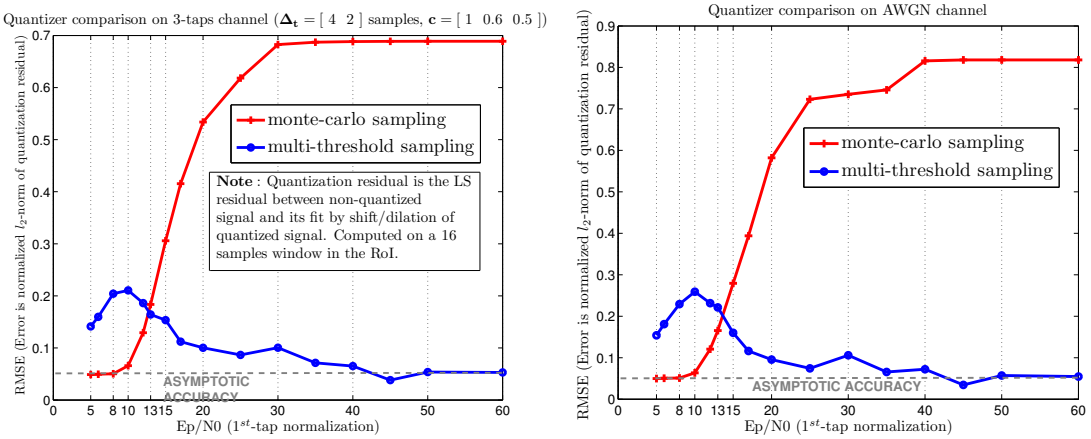
$$f_{\text{optimal}} = \arg\min_{f} \max_{\text{input } \boldsymbol{s}} \|f(\boldsymbol{\tau}, \boldsymbol{p}, \boldsymbol{R}) - \boldsymbol{s}\| . \tag{1.40}$$

It is a hard task, and the solution proposed is more heuristical than anything else. The estimated j^{th} sample of the signal is

$$\hat{s}_j = \begin{cases} \sum_i \tau_i \mathcal{N}\left(Q^{-1}(R_{i,j})\right) & ; \quad \exists R_{i,j} \neq 0,\ 1. \\ \tau_i,\ i = min_{\tilde{i} \in \{k:\ R_{k,j} - R_{k+1,j} \neq 0\}} \tilde{i} & ; \quad \text{otherwise.} \end{cases} \tag{1.41}$$

The rational is a rate estimate close to $\frac{1}{2}$ has a much lower variance than one close to 0 or 1. Thus a gaussian bell is applied to convert $Q^{-1}(R_{i,j})$ the estimated distance from the threshold into a score. For example, if one receives only 0s for a particular threshold, the estimated distance would be $+\infty$, however this is based on too few observations of a rare event. It makes more sense to compute a score taking into account the quality of the estimation.

One limitation is the threshold may not be tuned with the desired "finesse". It was taken into account and subsequent results use a tuning compatible with LP-BAN . Figure 1.12 shows the MT quantization fulfill its role at high SNR. Moreover best of both worlds can be combined. With this approach only the range 10-20dB is problematic. It may be improved using a different threshold than the default one for the MC algorithm.



(a) 3-taps channel (b) single tap

FIGURE 1.12: Quantization error comparison of MC and MT.

It is interesting both methods asymptotically reach a similar bound for a number of repetitions large enough. It would have been interesting to compare it with a uniform $log_2(P)$-bits quantizer to see if this bound is related to it.

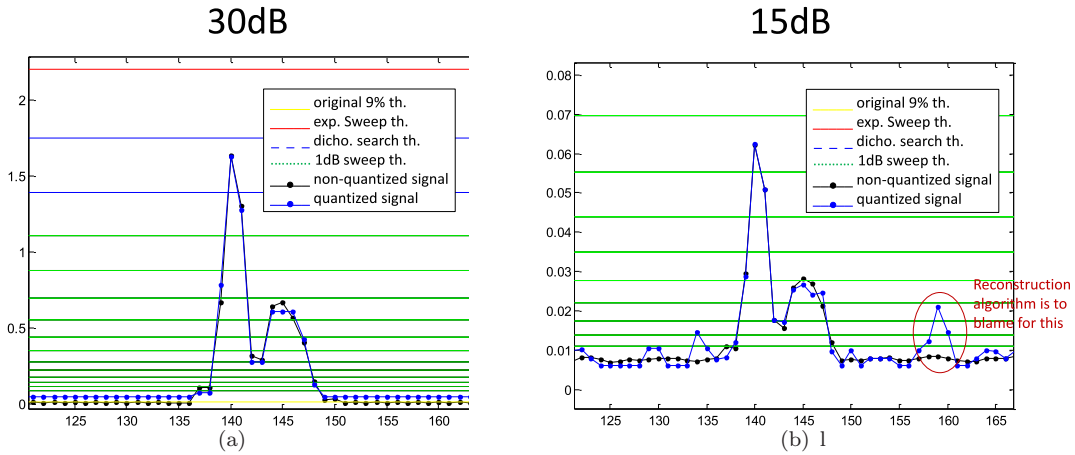


FIGURE 1.13: MT quantization at high SNR.

1.3.1.3 Hybrid solution

A simple hybrid solution can be built from the MC and MT algorithms:

- Apply the MT algorithm until $\tau_{\max}$ is found.
- If $\tau_{\max}/\tau_1$ is larger than some fixed constant, proceed with MT (current implementation uses 5dB).
- Else uses MC with threshold $\tau_{\max}/2$.

The resulting algorithm is listed in 1. It is by no mean optimal, and the number of levels in the "linear sweep" shall be reduced in order to minimize the variance of the estimator. From the LP-BAN specifications, a 5 levels uniform quantization was chosen. Call each of these levels $\tau_0 < \tau_1 < \cdots < \tau_5$. For a particular sample, if we pick one realization for each threshold, we obtain a binary vector like `[1 1 0 0 0]` or `[1 0 1 0 0]`. The first one is consistent and the second one is not since it shall be "above" τ_C and "below" τ_B. There are 6 consistent vectors : `[0 0 0 0 0]`, `[1 0 0 0 0]`, `[1 1 0 0 0]`, ..., `[1 1 1 1 1]`. One way to estimate the value of a sample, would be to combine realizations at random and consider only the consistent results. It can be achieved without the random combinations as the asymptotic result is a product of the rate of 0/1 for each threshold. Namely:

- to each consistent vector assign value v_i, $i = 0 \ldots 5$. In our case since levels are uniformely spaced: $v_i = i/5$
- for each sample estimate the probability p_i its true value s is above τ_i
- the estimated value for this sample is: $\hat{s} = \sum_{i=0}^{5} \left[v_i \prod_{j=0}^{i} p_j \prod_{k=i+1}^{5} (1 - p_k) \right]$.

This is the "Hybrid" quantization algorithm used in section 1.6. It is similar to algorithm 1 but for the linear sweep which has a fixed number of levels.

Algorithm 1 `quantize`: Hybrid quantization algorithm

Input: a number of repetions P and a repetition acquisition function `getRepetition()` returning an $1 \times N$ vector containing non-quantized samples,
τ_1 the original quantization threshold,
κ the minimum gain to enable MT quantization

Output: $\boldsymbol{X}_Q$ a $P \times N$matrix containing in each row $X_Q[i]$ the binary samples for the i^{th} repetition, $\boldsymbol{\tau}$ the threshholds for each repetition

textbfParameters $\boldsymbol{M}?[\dots M_i \dots]$, s.t. $M_i = \max_{M \in \mathbb{N}: \sum_{m=1}^{M}(2m-1) \leq P-i}$
$\max_d$ // t.b.d., maximum dichotomy depth
$H(\cdot)$ the Heaviside function.
n_c the number of consecutive all-0 quantized vector to confirm a threshold is too large.

$\boldsymbol{p} \leftarrow [11...1]$
$i \leftarrow 1$
$n \leftarrow 0$ // exponential sweep
while $j < n_c \wedge i < P$ **do**
  $X_Q[i] \leftarrow H(\texttt{getRepetition()} - \tau_i)$
  $j \leftarrow (X_Q[i] \neq [0\ 0\ \dots\ 0])?0 : j + 1$
  $\tau_{i+1} \leftarrow (X_Q[i] \neq [0\ 0\ \dots\ 0])?2\tau_i : \tau_i$
  $i++$
end while
$j \leftarrow 1$; step $\leftarrow \tau_i/2$; ld $\leftarrow i$; up $\leftarrow$ **false**; rm $\leftarrow 0$ // dichotomic search
while $M_j - \mathrm{rm} + i < P \wedge j < \mathrm{ld}, \max_d$ **do**
  step $\leftarrow$ step$/2$
  $\tau i \leftarrow \tau i - 1 +$ up?step : $-$step
  $X_Q[i] \leftarrow H(\texttt{getRepetition()} - \tau_i)$
  rm $\leftarrow$ rm $- (X_Q[i] = [0\ 0\ \dots\ 0])?0 : 2^j - 1$
  $i++; j++$
end while
if $\tau i - 1 < \kappa\tau_1$ **then**
  // fallback to MC
  while $i \leq P$ **do**
    $\tau_i \leftarrow \tau_{i-1} -$ step
    $X_Q[i] \leftarrow H(\texttt{getRepetition()} - \tau_i)$
    $i++$
  end while
else
  // linear sweep
  while $i \leq P$ **do**
    $\tau_i \leftarrow \tau_{i-1} -$ step
    if $\tau_i < \tau_1$ **then**
      $\tau_i \leftarrow \tau_{\mathrm{ld}-1}$
    end if
    $X_Q[i] \leftarrow H(\texttt{getRepetition()} - \tau_i)$
    $i++$
  end while
end if

1.3.2 Drift & Jitter

WARNING: Drift and jitter should not be understood in terms of the sampling clock, but of the time laps between two signal repetitions.

1.3.2.1 Drift

The drift D can be made relatively small if compensated, for the purpose of simulation, we will assume it is smaller than $\pm 10ppm$ (translated in term of the sampling clock), which means the signal will be shifted by at most 10 samples after 1 million samples acquired. The obvious effect, is the repetitions of the signal will not be combined coherently anymore. If we abstract the effect it may have on the MT quantization, the effect on the pulse shape amounts to a discrete convolution on a grid with interval $N \times T \times D$ – N the number of samples per pulse and T the sampling-step – with a box of width P the number of repetitions. So, the pulse is getting wider and its axis of symmtery is translated by $\frac{1}{2}P \times N \times D$. For a drift reasonnably small, the mismatch with the pulse-shape is expected to have little effect and the estimation of the location will be slightly biased. We observe such a behavior in LP-BAN with typical drift (section 1.6. The bias will be no more than 0.164 samples.

1.3.2.2 Jitter

We assume the jitter distribution is symmetrical. As P tends to infinity, the effect of jitter is to convolve the pulse shape with its distribution. The symmetry of the distribution makes the resulting pulse shape symmetrical as well. Figure 1.14 shows how given a jitter distribution, the pulse shape may vary. To overcome this effect, one may increase the number of repetitions P. However the price for this is to increase the effect of drift. One shall find a compromise between these two evils.

1.4 Estimating the number of taps

Two questions need to be answered:

1. Can the number of taps be reliably estimated before denoising?
2. How does performance varies with misestimation? especially shall one overestimate or underestimate the number of taps?

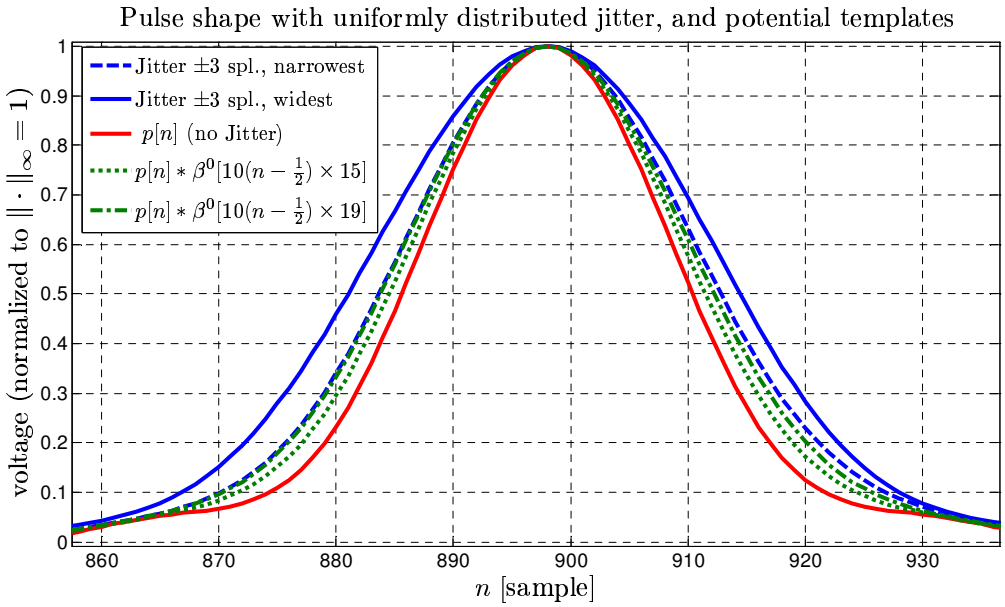


FIGURE 1.14: Effect of jitter on the pulse shape

A partial answer can be given to question 2. If the model is accurate, *i.e.* if the signal can be unambiguously described in term of it, then overestimation of the number of parameters is not a problem. In case it is not, which can be caused by excessive drift and jitter (with respect to the repetitions), there is no reliable way to cope with it. Indeed, preference may be granted to a simpler model even if it introduce a larger residual error. As the model mismatch grows, the line between a legitimate simpler explanation and missing some taps in the signal becomes thinner.

Since a relatively cheap algorithm will be introduced in chapter 2, we may indulge repeating denoising and the computation of the annihilating fitler. The following method is proposed:

- Start with a conservative number of taps. We use the minimum between a predifined maximum and the number of eigenvalues it takes to capture a certain energy in the denoising matrix (right now $1 - 10^{-4}$ of the total energy).
- From this starting point we decrease the estimated number of taps until we have a consistent result. A result is deemed *consistent* if estimated locations are separated by more than the minimum resolvable distance and estimated amplitudes are significantly above noise power and have a ratio greater than the minimum resolvable amplitude ratio (set to 1/5). The minimum resolvable distance is set to 1 sample, and amplitude limit is 4 times the estimated noise standard deviation.

- When a consistent result is obtained, we apply *Occam principle* by looking at a simpler model – 1 tap less – and see if the residual energy does not grow too much. As of today a simpler model is accepted if residual energy decreases.

This strategy leads to a very accurate estimate of the number of taps, and more important to good ranging performances. It is summarized in algorithm 4 along with the equalization.

1.5 Algorithmic summary

This sections contains the general description of the algorithm developped to solve the ranging problem. To complete the picture, quantization was outlined in 1 and fast Cadzow denoising algorithm will be developped in 2

Algorithm 2 `amplitudes`: Least-squares estimate of the taps amplitude

Input: $\boldsymbol{y}$ a vector of N (odd) samples,
$\boldsymbol{t}$ the estimated locations,
$p(\cdot)$ the pulse template (function).
Output: $\boldsymbol{c}$ the estimated amplitudes, optimal in the sense of least-square.

Compute w the maximum number of samples a single pulse "covers"
Find the set of samples $\mathcal{S} = \{s_i\} \leftarrow \bigcup_{t_k} [w \text{ samples closest to } t_k]$
Call $\boldsymbol{s}$ the elements of $\mathcal{S}$ organized in a column vector of length S.
Build a $S \times K$ matrix $\boldsymbol{A}$ s.t. $A_{j,k} \leftarrow p(s_j - t_k)$
return $\boldsymbol{c} \leftarrow \boldsymbol{A}^{\dagger}\boldsymbol{s}$, s.t. "†" denote the *Moore-Penrose* inverse (a.k.a. left pseudo-inverse)

Algorithm 3 `consistent`: Check the admissibility of a FRI estimation result

Input: $\boldsymbol{y}$ a vector of N (odd) samples,
$\boldsymbol{t}$ the estimated locations,
$p(\cdot)$ the pulse template (function).

Output: a boolean indicating consistency
// the following 3 constants may be modified
$\kappa \leftarrow 4$ // factor of how much a tap has to be above noise
$\rho \leftarrow \frac{1}{5}$ // fraction of the largest tap a tap has to be above
$\delta \leftarrow 1$ // minimum distance between taps (in samples)
if $\min(\Delta^{+}\boldsymbol{t}) < \delta \vee \min(\boldsymbol{c}) < \max(\kappa\sigma,\ \rho\max(\boldsymbol{c}))$ **then**
  return false
else
  return true
end if

Algorithm 4 `FRIanalysis`: FRI based signal analysis

Input: $\boldsymbol{y}$ a vector of N (odd) samples,
$K_{\max} \leq \frac{M-1}{2}$ the maximum number of taps expected,
$E_{\min}$ a lower bound estimate of the proportion of energy attributable to noise,
$\hat{\boldsymbol{p}}$ the DFT coefficients of a pulse template $\boldsymbol{p} = p(\lfloor N/2 \rfloor : \lfloor N/2 \rfloor)$, same dimension as $\boldsymbol{y}$,
M the (odd) number of DFT samples to run FRI on,
$\mu,\ \sigma$ the noise average and standard deviation.
Output: K the estimated number of taps,
$\boldsymbol{t} = [t_1,\ \ldots,\ t_K]$ and $\boldsymbol{c} = [c_1,\ \ldots,\ c_K]$ the locations and amplitudes of the taps.

$\hat{\boldsymbol{y}} \leftarrow \mathcal{F}\{\boldsymbol{y} - \boldsymbol{\mu}\}$
$\hat{\boldsymbol{y}}_{eq} \leftarrow \hat{\boldsymbol{y}} / \hat{\boldsymbol{p}}$ // element-wise division
$\boldsymbol{x} \leftarrow [\boldsymbol{x}_r\ \boldsymbol{x}_c] \leftarrow [[\hat{\boldsymbol{y}}_{eq}]_{1:\lceil M/2 \rceil}\ [\hat{\boldsymbol{y}}_{eq}]_{(N-\lceil M/2 \rceil):N}]$
$K \leftarrow \arg\min_{k=1\ldots K_{\max}} \texttt{lowrankApprox}(\texttt{toeplitz}(\boldsymbol{x}_r,\ \boldsymbol{x}_c))$// done during the first Cadzow iteration in practice
$\tilde{\boldsymbol{x}} \leftarrow \texttt{cadzow}(\boldsymbol{x}, K)$ // see chapter 2
$\boldsymbol{t} \leftarrow \texttt{roots}(\texttt{annihilatingFilter}(\tilde{\boldsymbol{x}}, K))$ // see [1]
$\boldsymbol{c} \leftarrow \texttt{amplitudes}(\boldsymbol{y},\ \boldsymbol{t}, p)$
// Loop until a consistent result is found
while $\neg\texttt{consistent}(\boldsymbol{c},\ \boldsymbol{t},\ \sigma) \wedge K > 1$ **do**
  $K \leftarrow K - 1$
  $\tilde{\boldsymbol{x}} \leftarrow \texttt{cadzow}(\boldsymbol{x}, K)$
  $\boldsymbol{t} \leftarrow \texttt{roots}(\texttt{annihilatingFilter}(\tilde{\boldsymbol{x}}, K))$
  $\boldsymbol{c} \leftarrow \texttt{amplitudes}(\boldsymbol{y},\ \boldsymbol{t}, p)$
end while
simpler $\leftarrow$ **true**
// Apply Occam principle: look for a simpler satisfying solution
$r_{\text{prev}} \leftarrow \left\| \boldsymbol{y} - \sum_{k=1}^{K} c_k \cdot p\left(\texttt{support}(\boldsymbol{y}) - t_k\right) \right\|$
while simpler $\wedge\ K > 1$ **do**
  $K \leftarrow K - 1$
  $\tilde{\boldsymbol{x}} \leftarrow \texttt{cadzow}(\boldsymbol{x}, K)$
  $\tilde{\boldsymbol{t}} \leftarrow \texttt{roots}(\texttt{annihilatingFilter}(\tilde{\boldsymbol{x}}, K))$
  $\tilde{\boldsymbol{c}} \leftarrow \texttt{amplitudes}(\boldsymbol{x},\ \tilde{\boldsymbol{t}}, \boldsymbol{p})$
  $r \leftarrow \left\| \boldsymbol{y} - \sum_{k=1}^{K} \tilde{c}_k \cdot p\left(\texttt{support}(\boldsymbol{y}) - \tilde{t}_k\right) \right\|$
  simpler $\leftarrow r_{\text{prev}} > r$ // may be tweaked.
  if simpler **then**
    $\boldsymbol{t},\ \boldsymbol{c}, r_{\text{prev}} \leftarrow \tilde{\boldsymbol{t}},\ \tilde{\boldsymbol{c}},\ r$
  else
    $K \leftarrow K + 1$
  end if
end while
return $K,\ \boldsymbol{t},\ \boldsymbol{c}$

1.6 Numerical results

1.6.1 Methodology

Simulations were performed with the equalized FRI algorithm on a 2-taps channel. Number of taps is estimated automatically as in algorithm 4. Several combinations of the following items are tested:

- Quantization: Infinite (`double` precision), 2-bits[3] and Hybrid (with 3 and 5 levels).
- Drift: none or $10ppm$.
- Jitter: none or uniformely distributed in ± 1.5 samples or ± 3 samples.

In order to better match the pulse shape when drift and jitter are present, we may use a wider template refered as *fat template*. It is obtained by convolution with a 3 samples wide box-function. This template may also be used with less jitter to see if a precise knowledge of the jitter distribution is necessary.

Each plot contains the RMSE and distribution of the error on the first location estimation. 50 trials were performed if not otherwise stated. To each color corresponds a different amplitude ratio between taps: 1/0 (no 2nd tap), 1/0.5 (2nd tap half of 1st tap), 1/1 (equal strength), 1/1.5, 1/2. Each page contains four plots with different distances between taps: 2, 3, 4 and 8 samples.

1.6.2 Analysis of the results

The error introduced by the cross-term is in general weaker than the one caused by drift, jitter or quantization. It is interesting to note that if no minimum distance between the paths is enforced (1.5 in current code), the algorithm will find an artificial tap corresponding to the cross-term for taps distants of 2 samples, thus reducing the error caused by the cross-term.

The algorithm can cope with drift and jitter efficiently thanks to a wider template. It does not require many bits of quantization as 2 seems to be enough. However, quantization with only 1bit which requires the use of a gain control quantization algorithm – like the hybrid algorithm – has drawbacks. It is not precise in a wide range, thus if the LoS pulse is significantly smaller than the largest tap (<1/2) it will probably be missed.

[3] we refer to a real 2-bit quantization, *i.e.* each sample acquired at a given time t may take one of four values

And this regardless of the SNR. The advantage of a parametric method as FRI over the traditional interpolation/maximum search disappear.

As a partial conclusion, it seems necessary to find a way to reliably quantize the signal with a large jitter in order to use a parametric method. In any other of the tested scenarii, it can be used to resolve close paths.

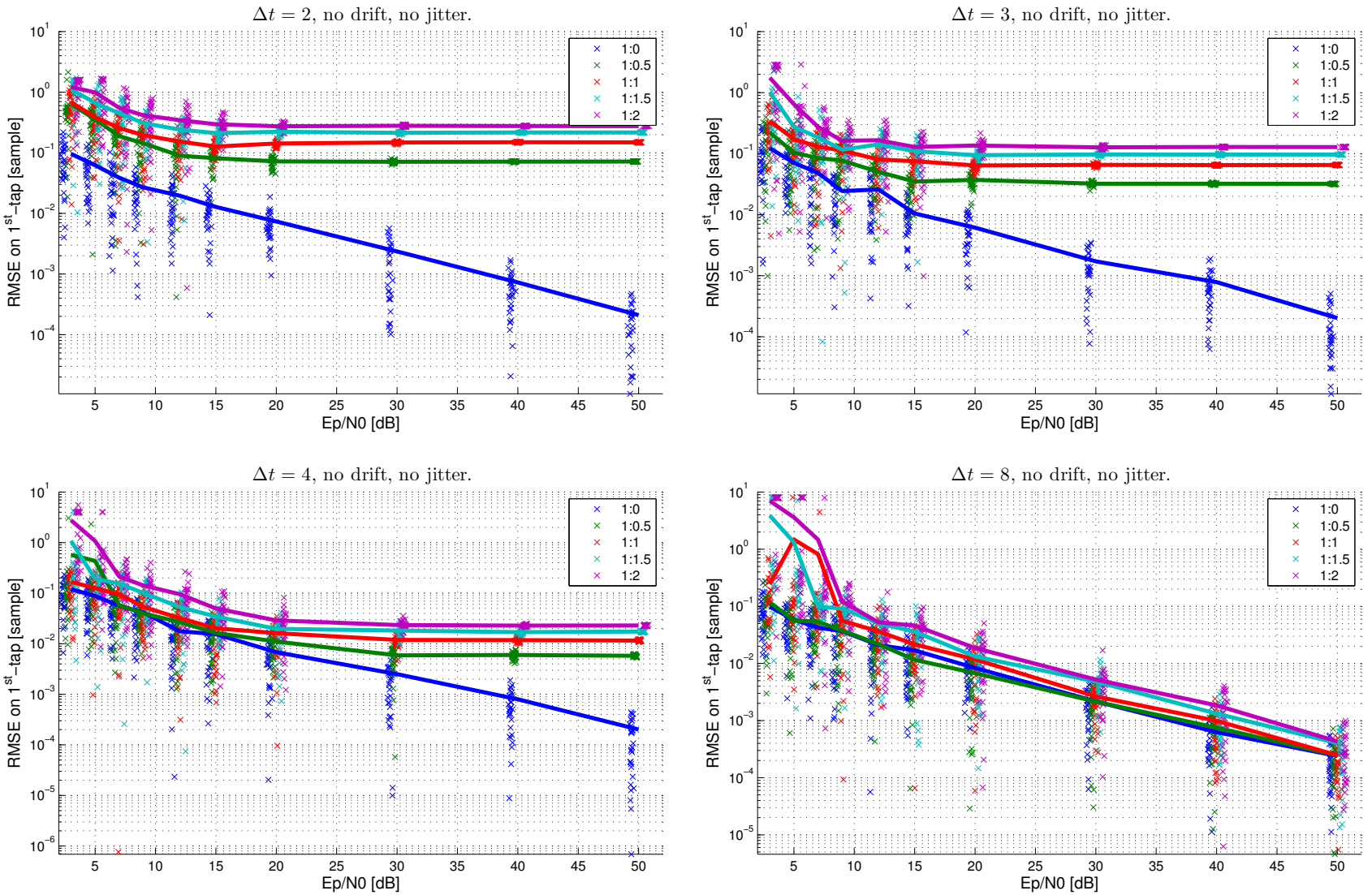


FIGURE 1.15: Infinite quantization (`double` precision) without drift nor jitter.

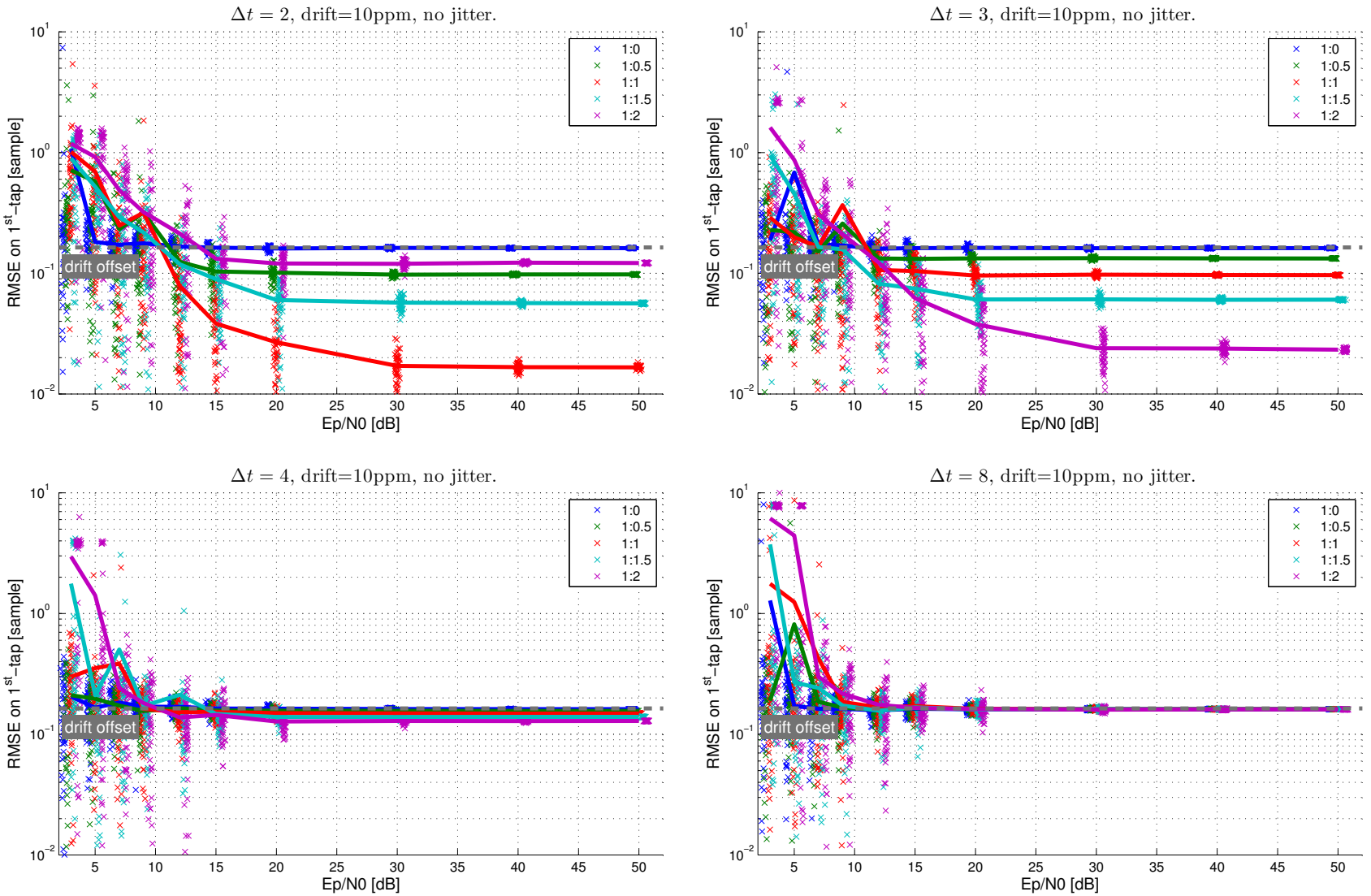


FIGURE 1.16: Infinite quantization (`double` precision) 10*ppm* of drift and no jitter.

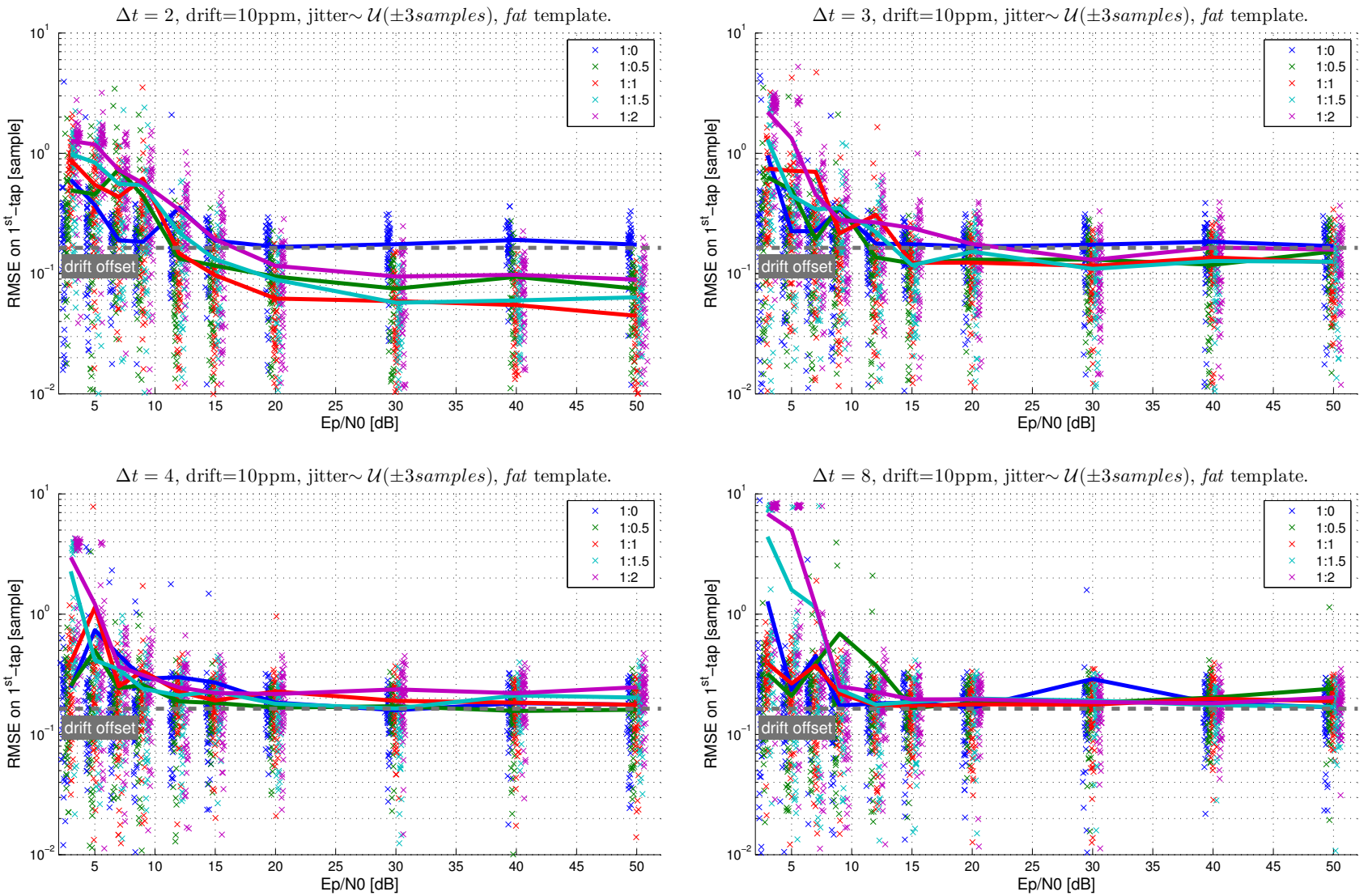


FIGURE 1.17: Infinite quantization (`double` precision) $10ppm$ of drift and ± 3 samples of jitter.

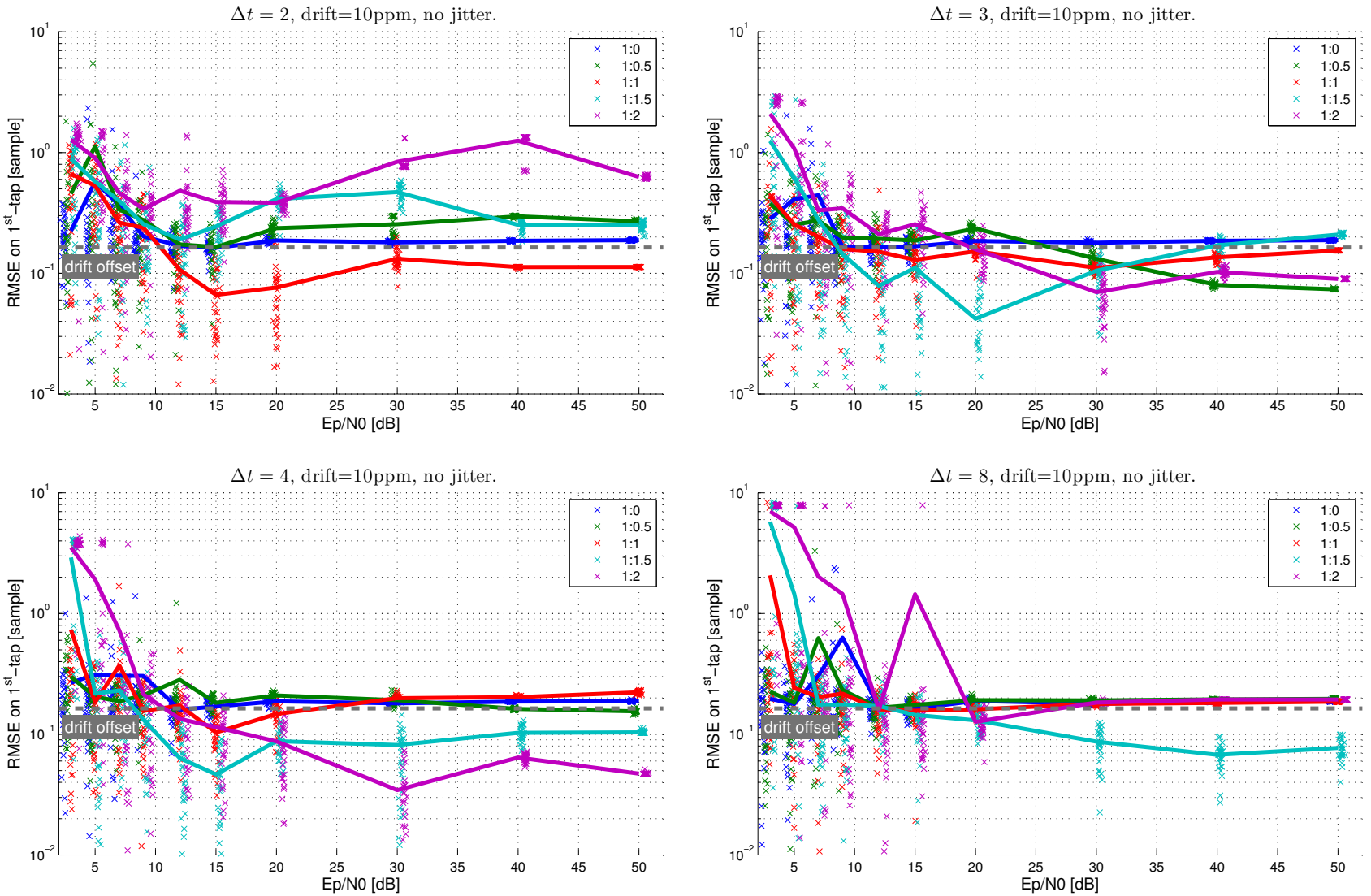


FIGURE 1.18: $2-bits$ quantization (`double` precision) $10ppm$ of drift and no jitter.

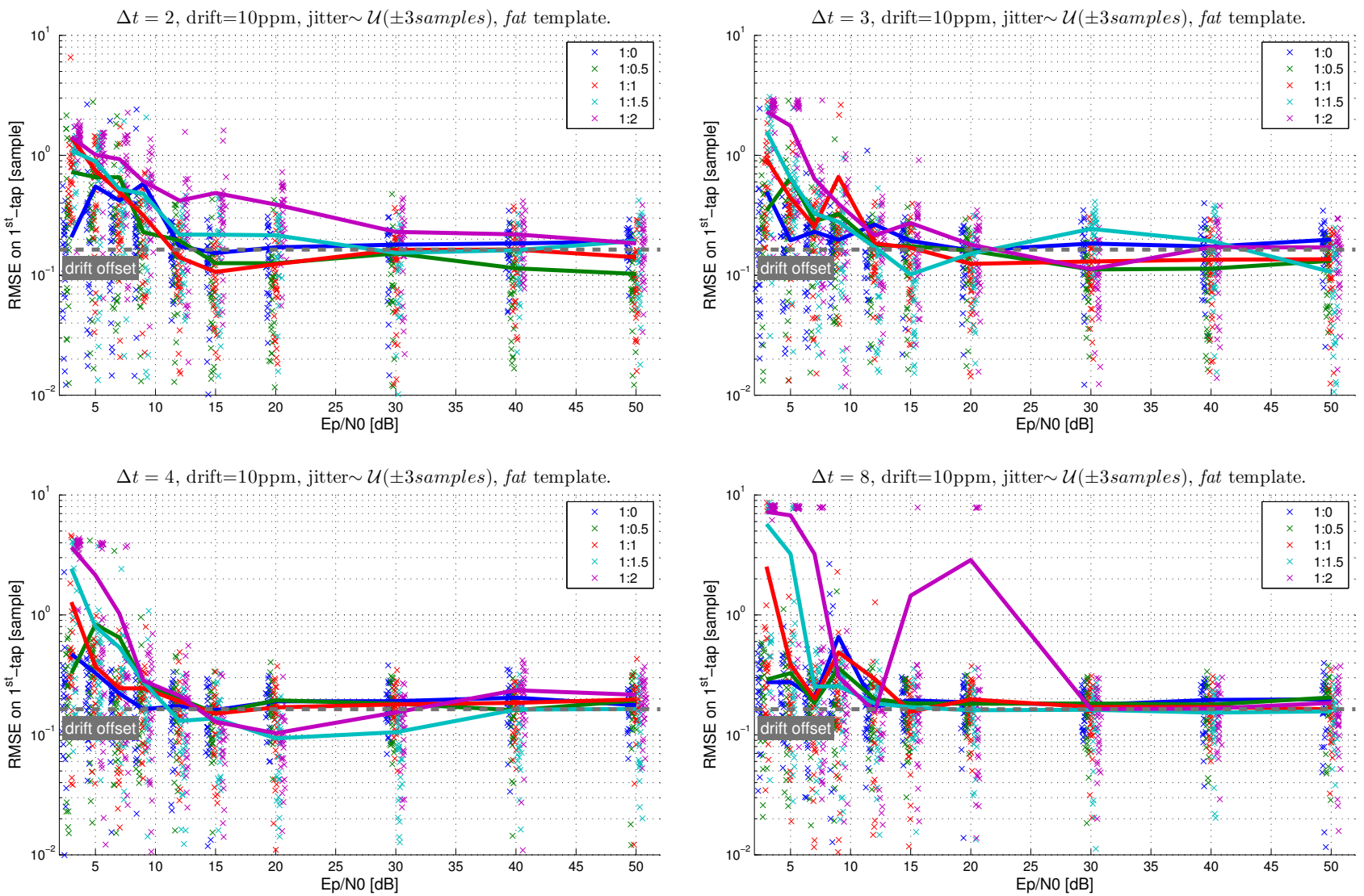


FIGURE 1.19: $2-bits$ quantization (`double` precision) $10ppm$ of drift and ± 3 samples of jitter.

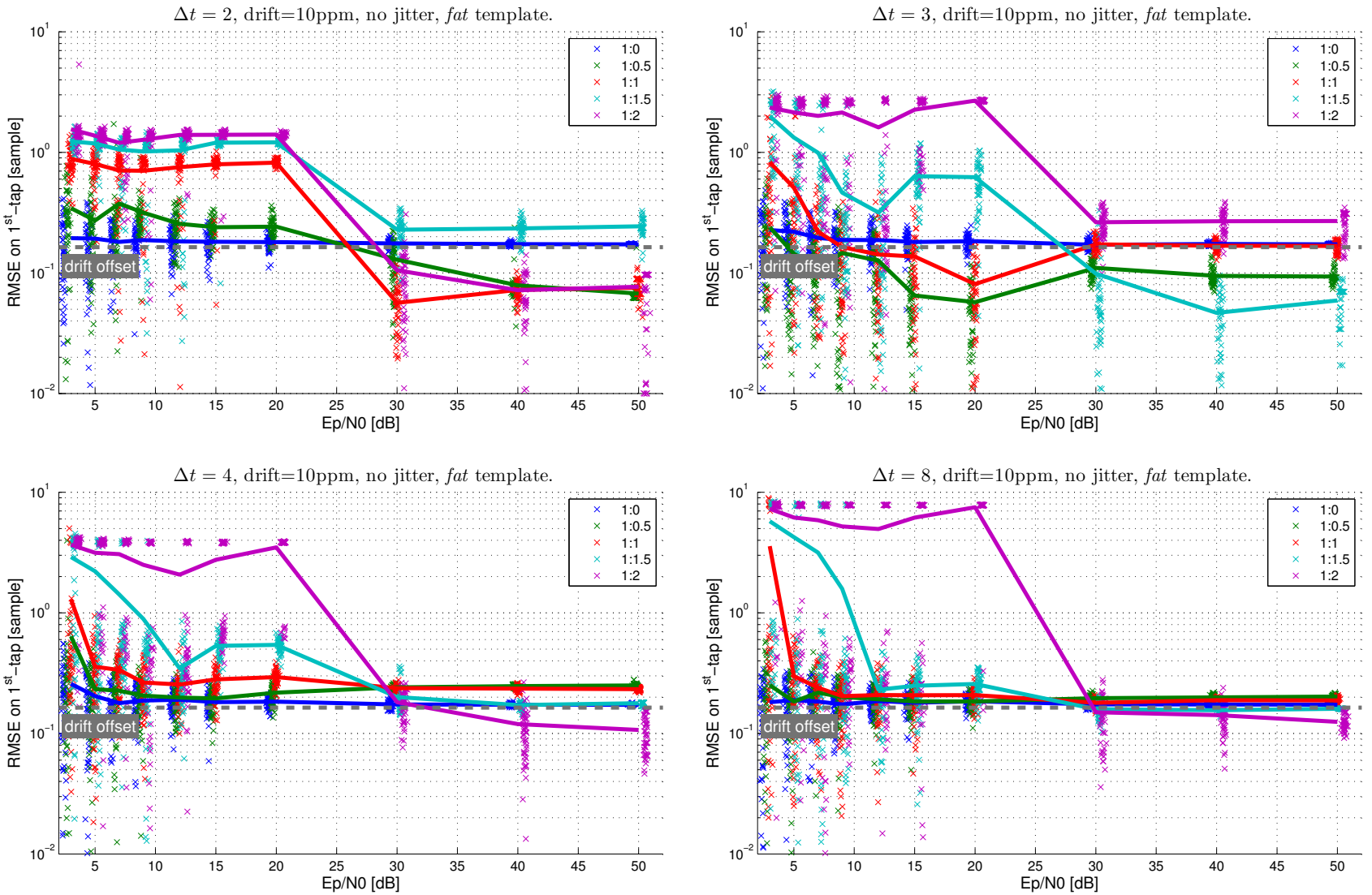


FIGURE 1.20: Hybrid quantization (`double` precision) 10*ppm* of drift and no jitter (uses a template slightly thinner than the *fat* template).

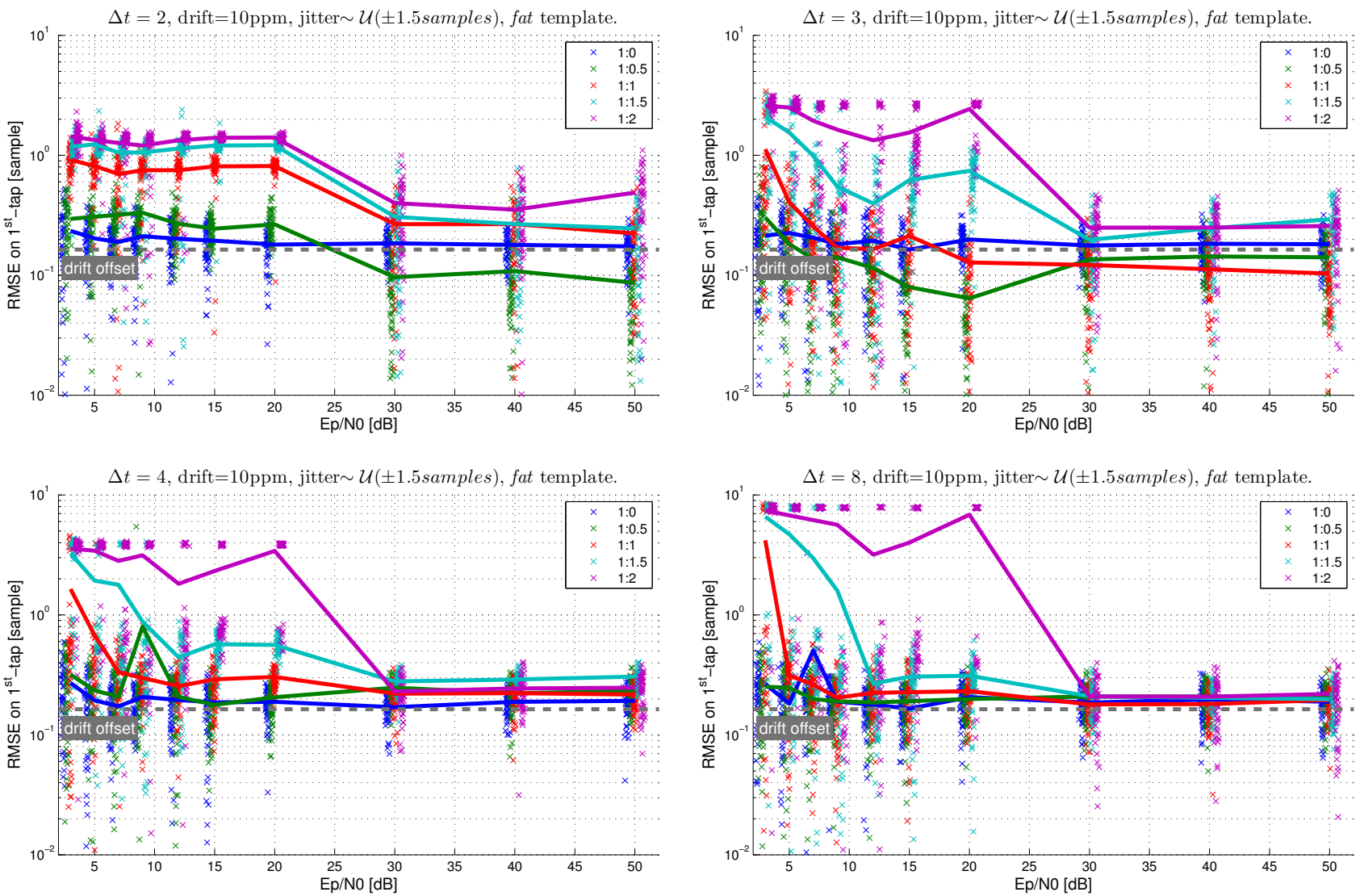


FIGURE 1.21: Hybrid quantization (`double` precision) $10ppm$ of drift and ± 1.5 samples of jitter.

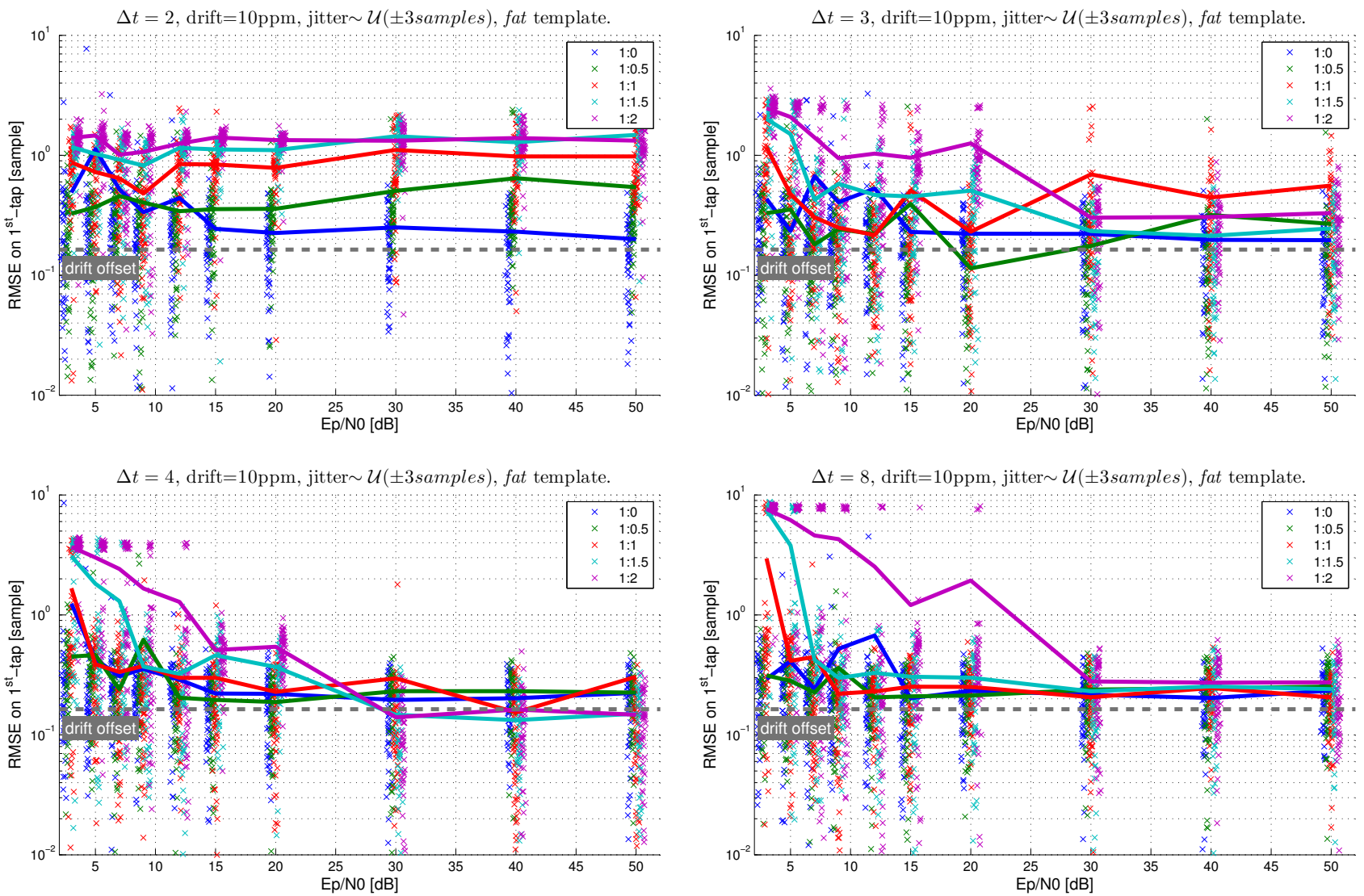


FIGURE 1.22: Hybrid quantization (`double` precision) $10ppm$ of drift and ± 3 samples of jitter.

Chapter 2

A faster denoising

2.1 Working with LP-BAN : Overview & Goals

The low-power and small form factor requirements of LP-BAN scream for a faster way to denoise the input signal. This chapter focuses on the Cadzow denoising method, and how it can be made faster for the problem at hand. The particularities of LP-BAN are:

- number of samples is small. Throughout this chapter we will use 31 as a relevant example.
- number of reflections is small. 3 reflections will be used as a typical example.
- computations done in fixed-point arithmetic.
- reduced set of fast built-in operations.
  - `+ - x`, boolean operations and register shifts are cheap
  - `/`, $\sqrt{\ }$ and trigonometric functions evaluations are costly
- additional hardware is limited by cost and space.

Each denoising iteration consist of finding a best low-rank approximation of a square toeplitz hermitian matrix – such a restriction holds given an odd number of samples. It can be achieved by keeping the largest spectral components. The obtention of these spectral components by an eigenvalue decomposition is the main contributor to the computational burden. More precisely, reducing the complexity of cadzow denoising boils down to an efficient solver for the partial symmetric eigenvalue problem. The principal eignevalues are supposed to be separated – *i.e.* distinguishable at machine precision.

The cost of trigonometric functions, divisions and square-roots rules out methods based on rotations like Jacobi procedures.([16] §8.6.3) The space and cost limitation makes 'divide and conquer' methods less attractive since they rely on paralellization to get high performances. Most of these 'unsuitable' algorithms have received a lot of attention for the past two decades as spectral analysis is usually performed with clusters of computers at hand rather than on embedded devices. This marketing analysis gives nevertheless a clue at which period we shall look for interesting litterature. Fixed-point arithmetic and reduced set of instructions hint at the 60s/70s.

The proposed method is an aggregate of well-established algorithms and a few more recent developpements. It can be summarized as a tridiagonalization followed by selective computation of eigenpairs. Choice is made to start with an expose of Rayleigh-Ritz algorithm and Krylov subspace methods, and then go on with the Lánczos iterations which are at the core of the algorithm. The reason for such a path is to emphasize Lánczos iterations are more than just a tridiagonalization tool. It is inspired and thus very similar to the treatment in Parlett's book.[17] The later proved to be an invaluable reference on the subject and is highly recommended to read. Note the algorithm was developped with LP-BAN 's characteristic in mind, however it is fairly general and potentially useful for other platforms as well.

The contribution of this chapter is to give a complete description of an efficient solver for the partial symmetric eigenproblem for small or large matrices, to provide a few propositions to glue the parts together, and some bounds to facilitate a fixed-point implementation.

2.2 The Rayleigh-Ritz algorithm and Krylov subspaces method

The reference with an outline the closest to this section is Parlett's book. [17] However the present expose aims at being self-contained, and some proofs are different – appropriate references are provided when similar. In addition it contains different concepts and some jargon, so for the sake of clarity, Figure 2.1 provides a schematic view of the progression.

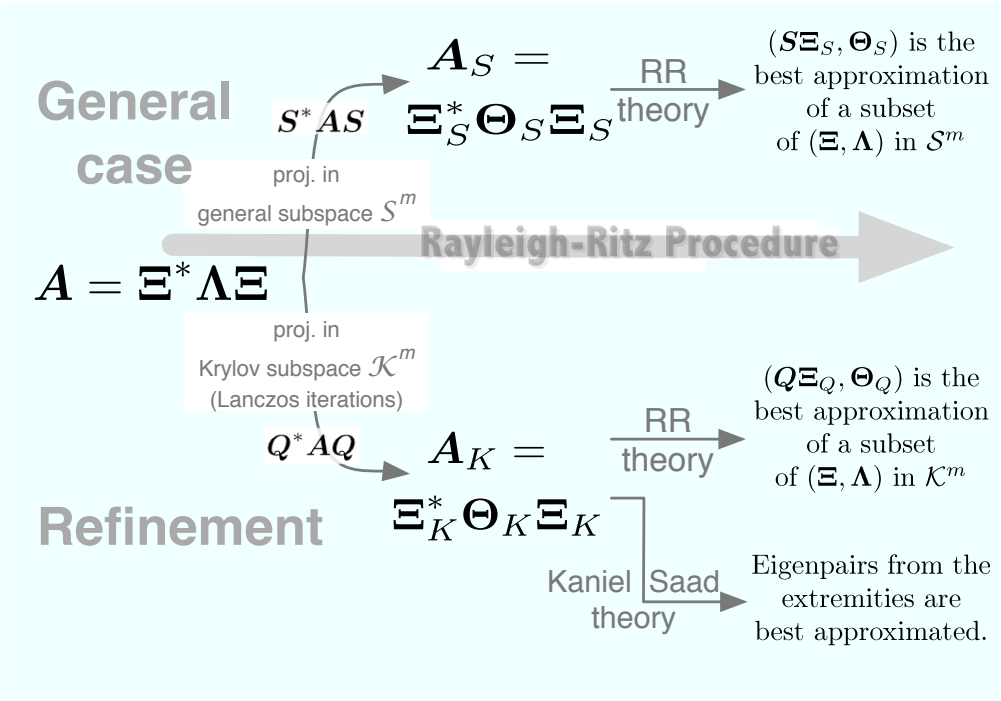


FIGURE 2.1: Overview of of Rayleigh-Ritz procedure and associates.

2.2.1 Eigenpairs approximation from a linear subspace: the Rayleigh-Ritz algorithm

Consider $\mathcal{Q}^m$ a linear subspace of $\mathbb{C}^n$ with dimension $m \leq n$. One way to define it is through an orthonormal basis Q, namely:

$$\mathcal{Q}^m = \text{span}(Q). \tag{2.1}$$

The question we are asking, is if there is a way to approximate some eigenpair of $\boldsymbol{A} \in \mathbb{C}^{n \times n}$ from a given subspace, and how can we do it.

Eigenvectors of $\boldsymbol{A}$ enjoy a fundamental property: they span a subspace which is *invariant with* A, *i.e.* if $\boldsymbol{\xi}$ is an eigenvector of $\boldsymbol{A}$, then $\boldsymbol{Ax} \in \text{span}(\boldsymbol{\xi})$, $\forall \boldsymbol{x} \in \text{span}(\boldsymbol{\xi})$. By linearity the same holds considering $\text{span}(\boldsymbol{\Xi})$, where $\boldsymbol{\Xi} = [\boldsymbol{\xi}_1, \ \ldots, \ \boldsymbol{\xi}_n]$.

If $\mathcal{Q}^m$ is not exactly invariant then the eigenvectors of $\boldsymbol{Q}\text{proj}_{\mathcal{Q}^m}\boldsymbol{A} = \boldsymbol{AQ}$ – called the Ritz vectors – match m eigenvectors of $\boldsymbol{A}$. However there is no hint at which is the index of the matched eigenvectors.

If $\mathcal{Q}^m$ is not invariant, an approximation of $\boldsymbol{A}$ eigenvectors may be derived via the *Rayleigh-Ritz* algorithm. It is a well-known and used procedure in the field of computational physics, civil engineering, ... In signal processing terminology, it is no more

than mapping the eigenpairs of $\text{proj}_{\mathcal{Q}^m}\boldsymbol{A} = \boldsymbol{Q}^*\boldsymbol{A}\boldsymbol{Q}$ by the inverse orthonormal transformation $\boldsymbol{Q}^*$. Fitness of the approximation is then measured in term of the residual error in the eigenvalue equation. Namely:

- Compute the projection $\tilde{\boldsymbol{A}} = \text{proj}_{\mathcal{Q}^m}\boldsymbol{A}$, called the Rayleigh quotient matrix by the physics folks.
- Compute $\left\{(\tilde{\lambda}_1,\ \tilde{\boldsymbol{\xi}}_1),\ \ldots,\ (\tilde{\lambda}_m,\ \tilde{\boldsymbol{\xi}}_m)\right\}$ the eigenpairs of $\tilde{\boldsymbol{A}}$.
- Compute the *Ritz pairs* of $\boldsymbol{A}$ from $\mathcal{Q}^m$: $(\theta_i,\ \boldsymbol{r}_i) = (\tilde{\lambda}_i,\ \boldsymbol{Q}\tilde{\boldsymbol{\xi}}_i)$.

An interesting property of the Ritz values is that they approximate well the eigenvalues up to the residual error in the eigenvalue equation [18]:

$$\text{For each Ritz pair } (\theta_i, \boldsymbol{r}_i), \exists\lambda \text{ an eigenvalue of } \boldsymbol{A} \text{ such that } |\lambda - \theta_i| \leq \|\boldsymbol{A}\boldsymbol{r}_i - \theta_i\boldsymbol{r}_i\|\,. \tag{2.2}$$

An obvious shortcoming of such a method is that several Ritz values can be matched to the same eigenvalue if their respective inequality overlap – rigorously said, if the solution intervals of the inequalities in 2.2 overlap. Moreover clustered eigenvalues will make it very likely to happen. Secondly, the fitness of the Ritz vectors cannot be precisely assessed generally. The exception is when eigenvalues are well separated. The *Gap theorem* precise this statement:

Theorem 2.1. ***[Parlett]***
Let λ be the closest eigenvalue of $\boldsymbol{A}$ to a Ritz value θ, and their associated eigen/Ritz pairs $(\lambda,\ \boldsymbol{\xi})$, $(\theta,\ \boldsymbol{r})$. Define the gap $\gamma = \min_{\lambda_i \neq \lambda} |\lambda_i - \theta|$, then

$$\sqrt{1 - \left(\frac{\|\boldsymbol{A}\boldsymbol{r} - \theta\boldsymbol{r}\|}{\gamma}\right)^2} \leq |\langle \boldsymbol{r}, \boldsymbol{\xi}\rangle| \quad (\leq 1). \tag{2.3}$$

and

$$|\lambda - \theta| \leq \frac{\|\boldsymbol{A}\boldsymbol{r} - \theta\boldsymbol{r}\|^2}{\gamma}. \tag{2.4}$$

Proof. See [17] §11.7. □

The culprits of the Rayleigh-Ritz algorithm for a *general* subspace $\mathcal{Q}^m$ are for us the lack of correspondance between indices of the spectrum and Ritz spectrum and the lack of extensibility. Lack of extensibility is failing to answer how to cheaply get a projection in a better subspace if not good enough – *i.e.* if inequalites overlap. A restriction to *Krylov subspaces* is aimed to answer both of these shortcomings.

2.2.2 Krylov subspaces reveal extremities of the spectrum

A basis for a dimension m *Krylov subspace* of the matrix (linear operator) $\boldsymbol{A}$ is obtained applying m-times $\boldsymbol{A}$ to a generating vector $\boldsymbol{f}$. This basis is denoted $K_{\boldsymbol{A}}^{m}(\boldsymbol{f}) = [\boldsymbol{f},\ \boldsymbol{A}\boldsymbol{f},\ \ldots,\ \boldsymbol{A}^{m-1}\boldsymbol{f}]$. The subspace itself is $\mathcal{K}_{\boldsymbol{A}}^{m}(\boldsymbol{f}) = \text{span}\,(K_{\boldsymbol{A}}^{m}(\boldsymbol{f}))$. Then any element $\boldsymbol{a}$ of this subspace has a natural polynomial representation:

$$\boldsymbol{a} = \sum_{i=0}^{m-1} a_i \cdot \boldsymbol{A}^i \boldsymbol{f} = \boldsymbol{f} p_{\boldsymbol{a}}(\boldsymbol{A}). \tag{2.5}$$

,where $p_{\boldsymbol{a}}(\cdot)$ is a polynomial of degree less than m.

This correspondence between Krylov subspaces and polynomial less than a certain degree yields some elegant and powerful results on the eigenpairs approximation. It may not be intuitive at first to build polynomials of a square matrix, however it gets easier to work with them as there is a correspondence between the *characteristic polynomial* of A and its *minimal polynomial*. The former is a polynomial of an argument in $\mathbb{C}$ while the later is for an argument in $\mathbb{C}^{m\times m}$. They are both defined over $\mathbb{C}$ so a priori a correspondence between them (may) make sense. We assume the notion of characteristic polynomial is well-known, and proceed with the definition of the minimal polynomial:

Definition 2.2. The minimal polynomial μ of $\boldsymbol{A} \in \mathbb{C}^{m\times m}$ is the monic polynomial over $\mathbb{C}$ of minimal degree having $\boldsymbol{A}$ as a root, *i.e.* $\mu(\boldsymbol{A}) = 0$.

Proposition 2.3. *Let $\boldsymbol{A} \in \mathbb{C}^{m\times m}$, p its characteristic polynomial, μ its minimal polynomial and λ one of its eigenvalues. Then:*

$$p(\lambda) = 0 \Leftrightarrow \mu(\lambda) = 0. \tag{2.6}$$

Proof.

$$\begin{aligned} \mu(\boldsymbol{A}) &= \sum_{i=0}^{n\leq m-1} \mu_i \boldsymbol{A}^i && (2.7)\\ &= \boldsymbol{\Xi}\left(\sum_{i=0}^{n\leq m-1} \mu_i \boldsymbol{\Lambda}^i\right)\boldsymbol{\Xi}^* \overset{def}{=} 0. && (2.8) \end{aligned}$$

Which in canonical form is equivalent to: $\prod_{i=0}^{n\leq m-1}(\boldsymbol{\Lambda} - \theta_i \mathbb{I}) = \boldsymbol{0}$. The polynomial of smallest degree satisfying this equation is $\prod_{\theta\in\mathcal{L}}(\boldsymbol{\Lambda} - \theta\mathbb{I})$. Thus:

$$\begin{aligned} \mu(\boldsymbol{A}) &= \boldsymbol{\Xi}\left(\prod_{\theta\in\mathcal{L}}(\boldsymbol{\Lambda} - \theta\mathbb{I})\right)\boldsymbol{\Xi}^* && (2.9)\\ &= \prod_{\theta\in\mathcal{L}}(\boldsymbol{A} - \theta\mathbb{I}). && (2.10) \end{aligned}$$

,with $\measuredangle$ the set of (unique) eigenvalues of $\boldsymbol{A}$. Thus p and μ have the same roots, with potentially different multiplicity. □

Put simply, μ is p with its duplicate canonical terms squashed. The above property is a stronger version of the Cayley-Hamilton theorem for square matrices which states μ divides p.

Corollary 2.4. ***[Parlett 1980]***
A vector $\boldsymbol{w} = \omega(\boldsymbol{A})\boldsymbol{f} \in \mathcal{K}_{\boldsymbol{A}}^{m}(\boldsymbol{f})$ *and* $(\theta, \boldsymbol{r})$ *a Ritz pair of the same Krylov subspace, then:*

$$\boldsymbol{w} \perp \boldsymbol{r} \quad \Leftrightarrow \quad \omega(\theta) = 0. \tag{2.11}$$

With this very handy definition for orthogonality, the Ritz vectors $\{\boldsymbol{r}_i\}_{i=1:m}$ obtained from $\mathcal{K}_{\boldsymbol{A}}^{m}(\boldsymbol{f})$ can be formulated as: Namely:

$$\boldsymbol{r}_i' = \underbrace{\prod_{\theta \in \measuredangle \setminus \{\theta_i\}} (\boldsymbol{A} - \theta\mathbb{I})}_{\mu_i(\boldsymbol{A})} \boldsymbol{f}, \quad \boldsymbol{r}_i = \boldsymbol{r}_i'/\|\boldsymbol{r}_i'\|. \tag{2.12}$$

It is simply the minimal polynomial with the canonical term corresponding to θ_i removed. Using this formulation of the Ritz vectors, Parlett derived the following bound:

Lemma 2.5. ***[Parlett 1980]*** *Define* $\varrho_{\boldsymbol{A}-\lambda_k\mathbb{I}}$ *the Rayleigh quotient of* $\boldsymbol{A} - \lambda_k\mathbb{I}$, $\boldsymbol{\Xi}_k = [\boldsymbol{\xi}_1 \cdots \boldsymbol{\xi}_k]$ *the orthonormal basis formed by its* k *principal eigenvectors and* $\boldsymbol{h}$ *the normalized orthogonal complement of* $\boldsymbol{t}$ *to* $\operatorname{span}(\boldsymbol{\Xi}_k)$*:*

$$\varrho_{\boldsymbol{A}-\lambda_k\mathbb{I}}(\pi(\boldsymbol{A})\boldsymbol{f}) \leq (\lambda_k^- - \lambda_n^-) \left[\frac{\sin[\arg(\boldsymbol{f}, \operatorname{span}(\boldsymbol{\Xi}_k))]}{\cos[\arg(\boldsymbol{f}, \boldsymbol{\xi}_k)]} \cdot \frac{\|\pi(\boldsymbol{A})\boldsymbol{f}\|}{\pi(\lambda_k)}\right]^2. \tag{2.13}$$

Proof. See [17]. □

Before going further, a useful theorem is needed. It will give the ability not only to derive bounds on the convegence of Ritz pairs to the two extremal eigenpairs of the spectrum but on their adjacent eigenpairs as well.

Theorem 2.6. ***[Courant-Fischer theorem]***
Let $\boldsymbol{A} \in \mathbb{C}^{n\times n}$ *an hermitian matrix and* $\varrho_{\boldsymbol{A}}(\boldsymbol{x}) = \frac{\boldsymbol{x}^*\boldsymbol{A}\boldsymbol{x}}{\langle \boldsymbol{x}, \boldsymbol{x}\rangle}$ *its Rayleigh quotient defined on* $\mathbb{C}^n$.

Consider λ_m^+ *the* m^{th} *eigenvalue of* $\boldsymbol{A}$ *in ascending order* ($\lambda_1^+ \leq \cdots \leq \lambda_n^+$) *and similarly* λ_m^- *in descending order. Then for any non-trivial* m*-dimensional linear subspace* $\mathcal{S}^m$ *of* $\mathbb{C}^n$

$$\lambda_m^+ = \min_{\mathcal{S}^m} \max_{\boldsymbol{x} \neq \boldsymbol{0} \in \mathcal{S}^m} \varrho_{\boldsymbol{A}}(\boldsymbol{x}). \tag{2.14}$$

and conversely

$$\lambda_m^- = \max_{\mathcal{S}^m} \min_{\boldsymbol{x}\neq\mathbf{0}\in\mathcal{S}^m} \varrho_{\boldsymbol{A}}(\boldsymbol{x}). \tag{2.15}$$

Proof. Proof of the converse: Define $\boldsymbol{\Xi}_m = [\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_m]$ the m-dimensional orthonormal basis formed with the m principal eigenvectors of $\boldsymbol{A}$.

$$\begin{aligned}\max_{\mathcal{S}^m} \min_{\boldsymbol{x}\neq\mathbf{0}\in\mathcal{S}^m} \varrho_{\boldsymbol{A}}(\boldsymbol{x}) &\geq \min_{\boldsymbol{x}\neq\mathbf{0}\in\mathrm{span}(\boldsymbol{\Xi}_k)} \varrho_{\boldsymbol{A}}(\boldsymbol{x}) \\ &= \boldsymbol{\xi}_m^* \boldsymbol{A} \boldsymbol{\xi}_m \;=\; \lambda_m^-.\end{aligned}$$

The intersection of a two subspaces of dimension m and $n-m+1$ both from a space of dimension n is necessarily non-trivial, *i.e.* it contains a non-0 element. Thus $\exists \boldsymbol{x} \neq \mathbf{0} \in \mathrm{span}([\boldsymbol{\xi}_m, \ldots, \boldsymbol{\xi}_n]) \cap \mathcal{S}^m$ which yields:

$$\begin{aligned}\min_{\boldsymbol{x}\neq\mathbf{0}\in\mathcal{S}^m} \varrho_{\boldsymbol{A}}(\boldsymbol{x}) &\leq \varrho_{\boldsymbol{A}}(\boldsymbol{x}) \\ &\leq \lambda_m^-.\end{aligned}$$

Thus $\lambda_m^- = \max_{\{\mathcal{S}^m\}} \min_{\boldsymbol{x}\neq\mathbf{0}\in\mathcal{S}^m} \varrho_{\boldsymbol{A}}(\boldsymbol{x})$. Proof for the min-max equation is similar. □

Corollary 2.7. ***[Cauchy interlace theorem]***

The Ritz values $\{\theta_1, \ldots, \theta_m\}$ *of any projection of* $\boldsymbol{A}$ *in a* m*-dimensional linear subspace verify:*

$$\lambda_k^+ \leq \theta_k^+ \leq \lambda_{n-m+k}^+ \tag{2.16}$$

$$\lambda_{n-m+k}^- \leq \theta_k^- \leq \lambda_k^- \tag{2.17}$$

The previous two results yield the following bounds for $\boldsymbol{x} \perp \mathrm{span}(\boldsymbol{\Xi}_{m-1})$:

$$0 \leq \theta_m^+ - \lambda_m^+ \leq \varrho_{\boldsymbol{A}-\lambda_m^+\mathbb{I}}(\boldsymbol{x}) \tag{2.18}$$

$$0 \leq \lambda_m^- - \theta_m^- \leq \varrho_{\boldsymbol{A}-\lambda_m^-\mathbb{I}}(\boldsymbol{x}) \tag{2.19}$$

The relationship between the minimal polynomial and the characteristic polynomial explicits $\boldsymbol{x} \perp (\mathrm{span}\boldsymbol{\Xi}_{k-1})$. Namely $P_{\boldsymbol{x}}(t)$ the polynomial representation of $\boldsymbol{x}$ must have roots matching the Ritz values of the subspace it is orthogonal to:

$$P_{\boldsymbol{x}}^{\perp}(t) = P(t) \prod_{i=1}^{k-1} (t - \theta_k) \tag{2.20}$$

Now the remaining task is to upper-bound as tightly as possible the ratio of polynoms in Parlett's inequality

$$\frac{\|P^{\perp}(\boldsymbol{A})\boldsymbol{h}\|}{|P^{\perp}(\lambda_k^-)|} \overset{2.20}{\leq} \frac{\|P(\boldsymbol{A})\boldsymbol{h}\|}{|P(\lambda_k^-)|}\prod_{i=1}^{k-1}\frac{\|\boldsymbol{A}-\theta_i^-\|}{|\lambda_k^- - \theta_i^-|} \tag{2.21}$$

$$\leq \frac{\|P(\boldsymbol{A})\boldsymbol{h}\|}{|P(\lambda_k^-)|}\prod_{i=1}^{k-1}\left|\frac{\theta_i^- - \lambda_n^-}{\theta_i^- - \lambda_k}\right| \tag{2.22}$$

Note $|\cdot|$ can be dropped around the product term as Corollary 2.7 guarantees positivity. Moreover:

$$\|P(\boldsymbol{A})\boldsymbol{h}\| = \left\|\boldsymbol{\Xi}\left(\sum_{i=0}^{n-k-1}\sigma_i\boldsymbol{\Lambda}^i\right)\boldsymbol{\Xi}^*\boldsymbol{h}\right\| \tag{2.23}$$

Since, $\boldsymbol{h} \perp \text{span}(\boldsymbol{\Xi}_k)$, one can write $\boldsymbol{\Xi}^*\boldsymbol{h} = [\boldsymbol{0}\boldsymbol{\xi}_{k+1}\cdots\boldsymbol{\xi}_n]^*\boldsymbol{h} \overset{def}{=} \boldsymbol{\Xi}_k^{\perp *}\boldsymbol{h}$. The polynomial part in equation 2.23 being diagonal:

$$\|P(\boldsymbol{A})\boldsymbol{h}\| = \left\|\boldsymbol{\Xi}_k^{\perp}\left(\prod_{i=0}^{n-k-1}p_i\mathbb{I} - \boldsymbol{\Lambda}\right)\boldsymbol{\Xi}_k^{\perp *}\boldsymbol{h}\right\|. \tag{2.24}$$

This rewriting is by no mean different from equation 2.23, it however makes clear one end of $\boldsymbol{A}$'s spectrum has no influence. Thus, from the maximization property of the eigenpairs:

$$\|P(\boldsymbol{A})\boldsymbol{h}\| \leq \max_{t\in[\lambda_n^- \ \lambda_{k+1}^-]}|P(t)|\,. \tag{2.25}$$

To maximize the ratio, one shall find a polynom as small as possible in the interval $[\lambda_n\lambda_{k+1}]$ and a very large magnitude at λ_k, these constraints are illustrated in 2.2(a). It is well-known[1] the polynom satisfying these conditions and maximizing the ratio is the *Chebyshev polynomial* of degree $m-k$, T_{m-k}. Figure 2.2(b) shows its remarkable properties. After scaling to map $[-1\ 1]$ to $[\lambda_n^- \ \lambda_{k+1}^-]$, the ratio evaluates to

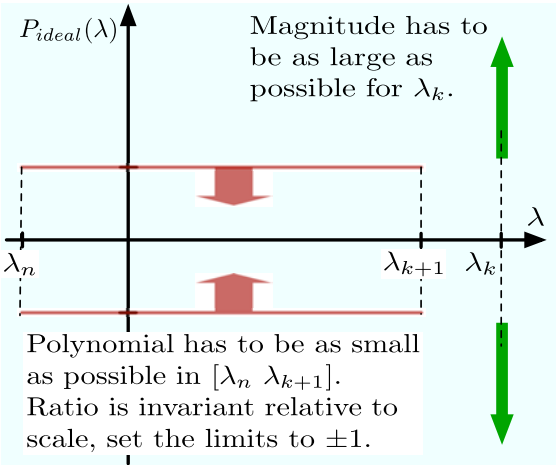
$$\min_{P\neq 0\in\mathbb{P}_{m-1}}\max_{t\in[\lambda_n^- \ \lambda_{k+1}^-]}\frac{\|P(t)\|}{|P(\lambda_k^-)|} = 1/T_{m-k}\left(1+2\frac{\lambda_k^- - \lambda_{k+1}^-}{\lambda_{k+1}^- - \lambda_n^-}\right). \tag{2.26}$$

It is clear the gap between λ_{k+1} and λ_k plays an important role in the tightness of the bound as it determines the "amount of absciss" the Chebyshev polynomial has, in order to grow above 1 – or in order for its inverse to get close to 0.

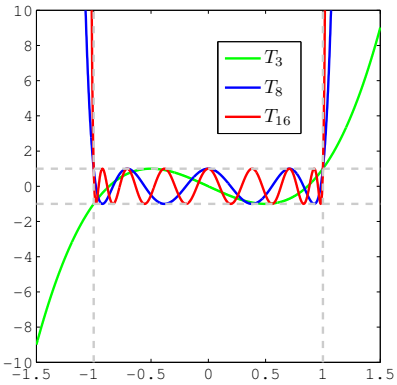
The inequation 2.13 combined with 2.16 now yield the Saad-Parlett bound:

Theorem 2.8. ***[Saad & Parlett 1980]***
Let $\boldsymbol{A} \in \mathbb{C}^{n\times n}$ with eigenpairs $\{(\lambda_i^-, \boldsymbol{\xi}_i^-)\}_{i=1:n}$ ordered by decreasing eigenvalue. The

[1]Translation: 'I am too lazy to write it down'. Look at [19] for good survey.



(a) Constraints on the "ideal" polynomial.



(b) Chebyshev polynomials are good candidates, in fact they are the best.

FIGURE 2.2: How to maximize the ratio of polynomials in Parlett's inequality 2.13

Ritz values $\{\theta_k^-\}_{k=1:m}$ *of* $\mathcal{K}_{\boldsymbol{A}}^m(\boldsymbol{f})$ *verify*

$$0 \leq \lambda_k^- - \theta_k^- \leq (\lambda_k^- - \lambda_n^-) \left[\frac{\sin[\arg(\boldsymbol{f}, \operatorname{span}(\boldsymbol{\Xi}_k))]}{\cos[\arg(\boldsymbol{f}, \boldsymbol{\xi}_k)]} \cdot \frac{\prod_{i=1}^{k-1} \frac{\theta_i^- - \lambda_n^-}{\theta_i^- - \lambda_k^-}}{T_{m-k}\left(1 + 2\frac{\lambda_k^- - \lambda_{k+1}^-}{\lambda_{k+1}^- - \lambda_n^-}\right)} \right]^2 . \tag{2.27}$$

and

$$\sin[\arg(\boldsymbol{\xi}_k, \mathcal{K}_{\boldsymbol{A}}^m(\boldsymbol{f}))] \leq \frac{\sin[\arg(\boldsymbol{f}, \operatorname{span}(\boldsymbol{\Xi}_k))]}{\cos[\arg(\boldsymbol{f}, \boldsymbol{\xi}_k)]} \cdot \frac{\prod_{i=1}^{k-1} \frac{\lambda_i^- - \lambda_n^-}{\lambda_i^- - \lambda_k^-}}{T_{m-k}\left(1 + 2\frac{\lambda_k^- - \lambda_{k+1}^-}{\lambda_{k+1}^- - \lambda_n^-}\right)}. \tag{2.28}$$

Proof. The above developement gives a flavor of the proof for 2.27. A complete proof is found in [17] §12.4 equation (12-4-1). □

Bounds from theorem 2.8 proved to be quite loose for $k > 1$. However, by the remarkable growth of the Chebyshev polynomial outside $[-1\ 1]$ witnessed in 2.2(b), the bound is in general tighter for a high degree polynomial, *i.e.* at the ends of the spectrum (a similar bound holds for the low end). It is important to state "in general" as the gap between an eigenvalue and its neighbors plays a role as well. In the Cadzow denoising problem, the largest leading eigenvalues are in general much better separated than the remaining ones as the later tend to cluster around 0. As well, the bound makes clear the choice of $\boldsymbol{f}$ is not crucial as long as it is not colinear to an eigenvector of $\boldsymbol{A}$. In the rest of the chapter we will thus drop $\boldsymbol{f}$ from the notation and suppose it was suitably selected.

Hence, Krylov subspaces are the tools of the trade for a quicker Cadzow algorithm. What is needed is an efficient algorithm to build an orthonormal projection into these subspaces.

2.3 Projection of an hermitian matrix into a Krylov subspace: the Lánczos iterations

The problem to be solved is the design an orthonormal basis $\boldsymbol{Q}_m$ such that $\boldsymbol{Q}_m^* \boldsymbol{A} \boldsymbol{Q}_m = \text{proj}_{\mathcal{K}^m(\boldsymbol{A})} \stackrel{def}{=} \boldsymbol{T}$. The obvious method is to generate m vectors by power iterations on a random $\boldsymbol{f}$, and then use an orthonormalization procedure like the Gram-Schmidt algorithm. However it is costly and it does not give insight on the nature of $\boldsymbol{T}$. *Lánczos iterations* [16, 20] perform the same task efficiently and the derivation of the algorithm reveals $\boldsymbol{T}$ is real, symmetric and tridiagonal. Before going further it is good to have a look at the algorithm itself.

Algorithm 5 Lánczos iterations ($\times m$)

Input: $\boldsymbol{A}$ an $n \times n$ hermitian matrix, $m \in \{1 : n\}$ the dimension of the Krylov subspace
Output: $\boldsymbol{Q}_m = [\boldsymbol{q}_1 \ \cdots \ \boldsymbol{q}_m]$ an $m \times m$ unitary matrix; $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ $n \times 1$ column vectors, such that $\boldsymbol{\alpha}$ the principal diagonal and $\boldsymbol{\beta}_{2:n}$ the first upper/lower diagonals of $\boldsymbol{T} = \boldsymbol{Q}_m^* \boldsymbol{A} \boldsymbol{Q}_m$.

```
pick an n × 1 vector r, possibly at random
set q_0 ← 0
for i = 1 : m do
    β_i ← ‖r‖
    q_i ← r/β_i
    r ← A q_i − β_i q_{i−1}
    α_i ← ⟨r, q_i⟩
    r ← r − α_i q_i
end for
return [q_1 ··· q_m], α, β
```

2.3.1 Derivation and properties of the Lánczos iterations

There is a beautiful way to derive the Lánczos iterations properties using the polynomial characterization of vectors in Krylov subspaces. The proof of the Lánczos algorithm given here relies solely on properties of orthogonal polynomials and elementary algebra. The connection between orthogonal polynomials and the Lánczos algorithm is of course well-known [20], however no proof based purely on orthonormal polynomial sequences and their properties were found in the litterature (if anybody knows of one, the author is interested).

FUN FACTS Cornelius Lánczos is an hungarian physicist, friend of Albert Einstein. He had to flew Europe as Nazi Germany started discriminating against jews. He landed in Purdue where divergent views with the dean of physics made him resigned. To make for a living he did some numerical analysis – subject he did not especially enjoy – for Boeing Inc. During this time he invented the FFT [21] – more than 20 years before Cooley & Tuckey rediscovered it independently. He invented the so-called *Lánczos Algorithm* in 1950 [20] before returning to physics in Ireland upon an offer from E. Schroedinger.

Definition 2.9. Sequence of orthogonal polynomials.
The sequence $p_0(t),\ p_1(t),\ \dots$ is a sequence of orthogonal polynomials *iif* $\forall i \geq 0$, p_i is a polynomial of degree i and $\langle p_i, p_{i+1}\rangle = 0$, where $\langle\cdot\rangle$ is an inner-product (maybe weighted).

Lemma 2.10. *([22] §22.1.4-5)*
A sequence of orthogonal polynomials admits a 3-terms recursion :

$$p_{i+1}(t) = (a_i t + b_i)p_i(t) + c_i p_{i-1}(t)\ , \quad \forall i > 0. \tag{2.29}$$

such that,

$$p_i(t) = k_i t^i + k_i' t^i + \dots\ , \quad b_i = \frac{k_{i+1}}{k_i}\ , \quad a_i = b_i\left(\frac{k_{i+1}'}{k_{i+1}} - \frac{k_i'}{k_i}\right)\ , \quad c_i = \frac{k_{i+1}k_{i-1}\langle p_i, p_i\rangle}{k_i^2\langle p_{i-1}, p_{i-1}\rangle}.$$

Proof. Assume $p_0(t),\ \dots,\ p_i(t)$ is an orthogonal sequence. Plugging in the coefficients one can verify $\langle p_{i+1}, p_i\rangle = \langle p_{i+1}, p_i\rangle = 0$. By linearity of the inner-product $\langle p_{i+1}, p_j\rangle = 0\ , \quad \forall j : 0 \leq j \leq i$. □

Each basis vector verifies $\boldsymbol{q}_i \in \mathcal{K}_{\boldsymbol{A}}^i$. Thus, the bijection between $\boldsymbol{q}_i \in \mathcal{K}_{\boldsymbol{A}}^i$ and $\mathbb{P}_{i-1}$ associated to the property $\langle \boldsymbol{q}_i, \boldsymbol{q}_j\rangle = \delta_{ij}$ together establish columns of $\boldsymbol{Q}_m$ form a sequence of orthogonal polynomials. From lemma 2.10 we deduce each of them can be computed from the previous two. Given the intitial conditions $\boldsymbol{q}_1 = \frac{\boldsymbol{f}}{\|\boldsymbol{f}\|}$ and

$\hat{\boldsymbol{q}}_2 = \boldsymbol{A}\boldsymbol{q}_1 - \text{proj}_{\boldsymbol{q}_1} A;\ \boldsymbol{q}_2 = \frac{\hat{\boldsymbol{q}_2}}{\|\hat{\boldsymbol{q}_2}\|}$:

$$\boldsymbol{q}_k = (c_{1,k}\boldsymbol{A} - c_{2,k}\mathbb{I})\boldsymbol{q}_{k-1} - c_{3,k}\boldsymbol{q}_{k-2}\ , \quad \forall k \in \{2:m\}. \tag{2.30}$$

The constants $c_{1,k},\ c_{2,k},\ c_{3,k}$ are to be determined. The orthogonality constraint yields:

$$\begin{aligned}
\langle \boldsymbol{q}_k, \boldsymbol{q}_{k-1}\rangle &= c_{1,k}\boldsymbol{q}_{k-1}^*\boldsymbol{A}\boldsymbol{q}_{k-1} - c_{2,k}\overbrace{\boldsymbol{q}_{k-1}^*\boldsymbol{q}_{k-1}}^{=1} - c_{3,k}\overbrace{\boldsymbol{q}_{k-1}^*\boldsymbol{q}_{k-2}}^{=0} \\
&= 0. \\
\Leftrightarrow \qquad c_{2,k} &= c_{1,k}\boldsymbol{q}_{k-1}^*\boldsymbol{A}\boldsymbol{q}_{k-1}.
\end{aligned}$$

and

$$\begin{aligned}
\langle \boldsymbol{q}_k, \boldsymbol{q}_{k-2}\rangle &= c_{1,k}\boldsymbol{q}_{k-2}^*\boldsymbol{A}\boldsymbol{q}_{k-1} - c_{2,k}\overbrace{\boldsymbol{q}_{k-2}^*\boldsymbol{q}_{k-1}}^{=0} - c_{3,k}\overbrace{\boldsymbol{q}_{k-2}^*\boldsymbol{q}_{k-2}}^{=1} \\
&= 0. \\
\Leftrightarrow \qquad c_{3,k} &= c_{1,k}\boldsymbol{q}_{k-2}^*\boldsymbol{A}\boldsymbol{q}_{k-1}.
\end{aligned}$$

This allows us to factor $c_{1,k}$ in the recursion equation:

$$c_{1,k}^{-1}\boldsymbol{q}_k \quad = \quad (\boldsymbol{A} - \underbrace{\boldsymbol{q}_{k-1}^*\boldsymbol{A}\boldsymbol{q}_{k-1}}_{\stackrel{def}{=}\ \alpha_{k-1}})\boldsymbol{q}_{k-1} - \underbrace{\boldsymbol{q}_{k-2}^*\boldsymbol{A}\boldsymbol{q}_{k-1}}_{\stackrel{def}{=}\ \beta_k}\boldsymbol{q}_{k-2} \stackrel{def}{=} \boldsymbol{r}_k. \tag{2.31}$$

It is natural to use the degree of freedom $c_{1,k}$ to ensure the normality of $\boldsymbol{q}_k$. Hence: $\gamma_k \stackrel{def}{=} c_{1,k}^{-1} = \|\boldsymbol{r}_k\|$. Obivously $\gamma_k,\ \alpha_k \in \mathbb{R}$. The former since it is a norm and the later since $\boldsymbol{A}$ is hermitian – indeed for a generic column vector $\boldsymbol{v}$, $[\boldsymbol{v}^*\boldsymbol{A}\boldsymbol{v}]^* = \boldsymbol{v}^*\boldsymbol{A}^*\boldsymbol{v} = \boldsymbol{v}^*\boldsymbol{A}\boldsymbol{v}$. In order to make the nature of the decomposed matrix clearer, equation 2.31 is rewritten as:

$$\boldsymbol{A}\boldsymbol{q}_{k-1} = \gamma_k\boldsymbol{q}_k + \alpha_k\boldsymbol{q}_{k-1} + \beta_k\boldsymbol{q}_{k-2}\ , \quad \forall k \in \{2:m\}. \tag{2.32}$$

which together whith the initial conditions is written in matrix notation as:

$$\boldsymbol{A}\boldsymbol{Q}_m = \boldsymbol{Q}_m\boldsymbol{T}_m \ , \quad \text{such that } \boldsymbol{Q}_m = [\boldsymbol{q}_1 \cdots \boldsymbol{q}_m] \tag{2.33}$$

$$\text{and } \boldsymbol{T}_m = \begin{pmatrix} \alpha_1 & \beta_2 & & & & & & \\ \gamma_1 & \alpha_2 & \ddots & & & & & \\ & \gamma_2 & \ddots & \ddots & & & \mathbf{0} & \\ & & \ddots & \ddots & \beta_k & & & \\ & & & \ddots & \alpha_k & \ddots & & \\ & \mathbf{0} & & & \gamma_k & \ddots & \ddots & \\ & & & & & \ddots & \ddots & \beta_m \\ & & & & & & \gamma_{m-1} & \alpha_m \end{pmatrix} .$$

Moreover $\boldsymbol{T}_m$ is hermitian as $\boldsymbol{T}_m^* = [\boldsymbol{Q}_m^*\boldsymbol{A}\boldsymbol{Q}_m]^* = \boldsymbol{Q}_m^*\boldsymbol{A}^*\boldsymbol{Q}_m = \boldsymbol{Q}_m^*\boldsymbol{A}\boldsymbol{Q}_m = \boldsymbol{T}_m$. Thus $\beta_k = \gamma_{k-1}^* = \|\boldsymbol{r}_k\| \ , \quad \forall k \in \{2 : m\}$, which makes $\boldsymbol{T}_m$ a real, symmetric tridiagonal matrix:

$$\boldsymbol{T}_m = \begin{pmatrix} \alpha_1 & \beta_2 & & & & & & \\ \beta_2 & \alpha_2 & \ddots & & & & & \\ & \beta_3 & \ddots & \ddots & & & \mathbf{0} & \\ & & \ddots & \ddots & \beta_k & & & \\ & & & \ddots & \alpha_k & \ddots & & \\ & \mathbf{0} & & & \beta_{k+1} & \ddots & \ddots & \\ & & & & & \ddots & \ddots & \beta_m \\ & & & & & & \beta_m & \alpha_m \end{pmatrix} . \tag{2.34}$$

To summarize: $\boldsymbol{r}_k = \boldsymbol{A}\boldsymbol{q}_{k-1} - \alpha_k\boldsymbol{q}_{k-1} - \beta_k\boldsymbol{q}_{k-2}$, $\alpha_k = \boldsymbol{q}_{k-1}^*\boldsymbol{A}\boldsymbol{q}_{k-1}$, $\beta_k = \|\boldsymbol{r}_k\|$, $\boldsymbol{q}_k = \frac{\boldsymbol{r}}{\|\boldsymbol{r}_k\|}$, which is exactly one step of algorithm 5. Note that the algorithm performs slightly different but equivalent operations. It has been shown to be numerically the most stable implementation [23]. We proceed with a very useful lemma on the roots of polynomials forming an orthonormal sequence:

Lemma 2.11.

Roots of polynomials p_i and p_{i+1} from an orthogonal sequence strictly interlace. $\{p_i\}$ is called a Sturm sequence.

Proof. See Szegő[24] §3.2 or Chihara [19] §5. □

Corollary 2.12. ***[Sturm sequence property]***

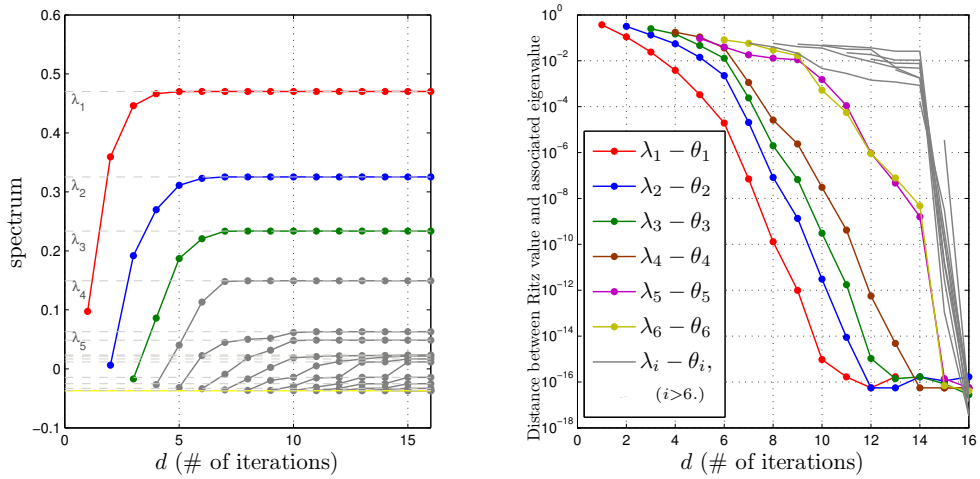
The eigenvalues of a real tridiagonal symmetric $\boldsymbol{T}_{i+1}$ *obtained by Lánczos algorithm and* $\boldsymbol{T}_i \overset{def}{=} \boldsymbol{T}_{i+1;1:i,1:i}$ *strictly interlace.*

Proof. By the polymial characterization of vectors in a Krylov subspace, $\boldsymbol{T}_{i+1}$ and $\boldsymbol{T}_i$ can be represented by their minimal polynomial. Since the minimal polynomial divides the characteristic polynomial, each root of the minimal polynomial is an eigenvalue (no multiple eigenvalue). By construction, the minimal polynomials of $\boldsymbol{T}_j$ and $\boldsymbol{T}_i$ are orthogonal $\forall j \neq i$. Applying previous lemma proves the claim. □

Corrolary 2.12 will later come in handy to compute a few eigenvalues of $\boldsymbol{T}_m$.

2.3.2 Krylov subspace projection for LP-BAN

It is time to show the above method can be fruitfuly used to denoise LP-BAN data. The experiment was made of 31 equalized FFT samples from the LP-BAN simulator, arranged in a 16×16 hermitian toeplitz matrix $\boldsymbol{A} = \boldsymbol{\Xi}^* \boldsymbol{\Lambda} \boldsymbol{\Xi}$. The signal contained 3 pulses. Projection of $\boldsymbol{A}$ in $\mathcal{K}^d(\boldsymbol{A}; \boldsymbol{f}) = \boldsymbol{Q}_d^* \boldsymbol{T}_d \boldsymbol{Q}_d$ was computed by Lánczos iterations with PRO, for $d = 3, \ldots, m$. Define $\boldsymbol{T}_d = \boldsymbol{\Xi}_{K,d}^* \boldsymbol{\Theta}_d \boldsymbol{\Xi}_{K,d}^*$ the eigenvalue decomposition of $\boldsymbol{T}_d$. All eigenvalue decompositions were performed with MATLAB `eig` function. Figure



(a) Ritz values converge to outermost eigenvalues first. The positive end of the spectrum attracts the Ritz values as its magnitude is much larger than the one of the negative end.

(b) distance between Ritz values and their closest eigenvalue. (log_{10} scale)

FIGURE 2.3: Convergence of outermost Ritz values.

2.3 shows the convergence of Ritz values to outermost eigenvalues.

Table 2.1 verifies the ritz spectrum obtained by full Lánczos process (tridiagonalization of $\boldsymbol{A}$) matches the spectrum of $\boldsymbol{A}$. It shows PRO is effective to counter loss of orthogonality.

Table 2.2 shows convergence of the 3 principals Ritz pairs obtained by partial Lánczos process with increasing number of steps to the 3 principals eigenpairs obtained by full Lánczos process. It shows Kaniel-Saad theory can be successfuly applied to reduce computations.

$\lvert\Lambda-\Theta_{16}\rvert$	$\lvert\,\lvert(\boldsymbol{Q}_{16}\boldsymbol{\Xi}_{K,16})^*\boldsymbol{\Xi}\rvert-\mathbb{I}\rvert$								
0	7.E-10	2.E-10	7.E-10	5.E-10	$\cdots$	7.E-10	7.E-10	6.E-10	6.E-10
2.08E-17	3.E-09	7.E-10	3.E-09	2.E-09	$\cdots$	3.E-09	3.E-09	3.E-09	3.E-09
1.39E-17	1.E-11	3.E-12	1.E-11	9.E-12	$\cdots$	1.E-11	1.E-11	9.E-12	9.E-12
1.39E-17	7.E-13	2.E-13	7.E-13	5.E-13	$\cdots$	3.E-12	5.E-12	6.E-12	1.E-11
3.47E-18	1.E-12	3.E-13	2.E-12	1.E-12	$\cdots$	5.E-12	8.E-12	9.E-12	1.E-11
3.47E-18	4.E-13	8.E-14	4.E-13	3.E-13	$\cdots$	2.E-12	3.E-12	3.E-12	3.E-12
5.03E-17	5.E-15	1.E-15	7.E-16	5.E-14	col.	3.E-12	5.E-12	5.E-12	8.E-12
6.25E-17	5.E-13	1.E-13	5.E-13	3.E-13	5 to 12	9.E-13	2.E-12	2.E-12	3.E-12
4.86E-17	2.E-13	4.E-14	2.E-13	2.E-13	skipped	1.E-12	2.E-12	2.E-12	3.E-12
3.47E-18	5.E-13	1.E-13	5.E-13	3.E-13	$\cdots$	1.E-12	3.E-12	3.E-12	5.E-12
4.16E-17	3.E-14	7.E-15	3.E-14	3.E-14	$\cdots$	3.E-13	5.E-13	5.E-13	7.E-14
5.55E-17	7.E-14	2.E-14	7.E-14	6.E-14	$\cdots$	4.E-13	6.E-13	5.E-13	1.E-12
5.55E-17	7.E-17	2.E-16	3.E-16	1.E-17	$\cdots$	3.E-16	2.E-15	2.E-15	7.E-14
2.78E-17	2.E-16	2.E-16	2.E-16	1.E-16	$\cdots$	7.E-16	2.E-16	1.E-15	2.E-14
1.67E-16	3.E-16	4.E-17	2.E-16	1.E-16	$\cdots$	4.E-16	8.E-16	7.E-16	5.E-15
5.55E-17	3.E-16	2.E-17	2.E-16	2.E-16	$\cdots$	2.E-16	7.E-18	2.E-16	2.E-15

TABLE 2.1: Accuracy of full-Lánczos with PRO

d	$\lvert\lambda_i-\theta_{d,i}\rvert,\ i=1,2,3.$			$\lvert1-(\boldsymbol{Q}_{16}\xi_{16,i})^*\boldsymbol{Q}_d\xi_{d,i}\rvert,\ i=1,2,3.$		
3	0.024105	0.133501	0.250646	0.066269	0.609642	0.959551
4	0.003834	0.055381	0.147553	0.010151	0.295366	0.814299
5	0.000335	0.014201	0.046684	0.00068	0.063985	0.316324
6	1.94E-05	0.002273	0.012965	2.76E-05	0.006373	0.057887
7	7.07E-08	2.06E-05	0.000241	7.89E-08	3.44E-05	0.000591
8	1.30E-10	8.24E-08	2.00E-06	1.45E-10	1.38E-07	4.94E-06
9	9.83E-13	1.35E-09	6.66E-08	1.10E-12	2.29E-09	1.67E-07
10	9.44E-16	3.05E-12	3.06E-10	4.44E-16	4.82E-12	6.87E-10
11	1.67E-16	8.83E-15	1.73E-12	2.22E-15	1.47E-14	3.50E-12
12	5.55E-17	5.55E-17	1.05E-15	2.00E-15	1.67E-15	2.66E-15
13	1.67E-16	5.55E-17	1.39E-16	2.00E-15	2.00E-15	8.88E-16
14	0	1.67E-16	1.67E-16	1.78E-15	1.67E-15	5.55E-16
15	5.55E-17	1.11E-16	8.33E-17	1.55E-15	2.44E-15	1.44E-15
16	5.55E-17	1.67E-16	2.78E-17	2.66E-15	1.67E-15	3.33E-16

TABLE 2.2: Convergence of outer Ritz pairs to corresponding eigenpairs.

2.3.3 Lánczos algorithm in finite-precision arithmetic

It was known by Lánczos himself that his algorithm suffers from numerical instability in finite-precision arithmetic. Later developements by Paige [25] show a lethal loss of

orthogonality between the basis vectors happens whenever a Ritz pair has converged to a corresponding eigenpair. Moreover the error on the basis vector is in the direction of the converged Ritz vectors. It has the devastating effect to *clone* the converged Ritz value. This artifact is usually refered as the *ghost eigenvalue problem* in the litterature and is illustrated in table 2.3.

Spectrum of		
$\boldsymbol{A}$	$\boldsymbol{T}_{16}$ w/ PRO	$\boldsymbol{T}_{16}$ (plain)
0.4703,	0.4703,	***0.4703***
0.3253,	0.3253,	***0.4703***
0.2336,	0.2336,	0.3253
0.1493,	0.1493,	0.2336
0.0631,	0.0631,	0.1493
0.0486,	0.0486,	0.0631
0.0236,	0.0236,	0.0486
0.0217,	0.0217,	0.0236
0.0167,	0.0167,	0.0217
0.0118,	0.0118,	0.0167
-0.0145,	-0.0145,	0.0118
-0.0252,	-0.0252,	-0.0145
-0.0330,	-0.0330,	-0.0252
-0.0369,	-0.0369,	-0.033
-0.0372,	-0.0372,	-0.0369
-0.0372,	-0.0372,	-0.0372

TABLE 2.3: The ghost eigenvalue problem and a possible solution

Several fixes have been proposed. The soundest of all was proposed by Parlett ([17] §13.8) and is called *selective orthogonalization (SO)*. It tracks convergence of Ritz pairs at each step i by performing an eigenvalue decomposition of $\boldsymbol{T}_i$. If new pair(s) have converged it orthogonalize the last basis vector against all the converged ritz vectors. It proved to be a very efficient and relatively economic procedure.

Another procedure – which we will use – is the *partial reorthogonalization (PRO)* of Simon.[23] It estimates the loss of orthogonality at each step and reorthogonalize the new basis vector against the ones with an inner-product crossing a threshold ν_{low} and their adjacent basis vectors ν_{hi}. The hysteresis threshold (ν_{low}, ν_{hi}) was set by statistical anlysis and then backed by some analytical arguments.

It is not in the scope of this report to analyse reorthogonalization procedures. As an end note, PRO was chosen over SO for its relative simplicity. Further analysis would be required to make a more educated choice.

2.3.4 A practical stopping criterion

It may not be apparent from theorem 2.8 that Ritz vectors converge at a speed similar to the Ritz values. To show it, given the bound on the Ritz value $\alpha_k^- - \theta_k^- \leq (\alpha_k^- - \alpha_n^-)\epsilon^2$:

$$\sin[\arg(\boldsymbol{\xi}_k, \mathcal{K}_{\boldsymbol{A}}^m(\boldsymbol{f}))]^2 \leq \epsilon^2 \tag{2.35}$$

$$1 - \langle \boldsymbol{\xi}_k, \boldsymbol{\psi} \rangle \langle \boldsymbol{\xi}_k, \boldsymbol{\psi} \rangle^* = \tag{2.36}$$

$$1 - |\langle \boldsymbol{\xi}_k, \boldsymbol{\psi} \rangle| \leq \tag{2.37}$$

where $\boldsymbol{\psi}$ is the closest unit-norm vector in $\mathcal{K}_{\boldsymbol{A}}^m$ to $\boldsymbol{\xi}_k$. It tells us the inner-product between Ritz and eigen vectors converge to 1 at a speed comparable to the Ritz/eigen values. This is backed up by table 2.2, and basing the stopping criterion on the innermost eigenvalue to be computed seems reasonnable. The idea is to use the monotonic convergence of the Ritz values from below. It is guaranteed by theorem 2.7. It is additionally supposed convergence speed is decreasing, which is observed in simulations – however it is not mandatory. Let the innermost Ritz value have index K (decreasing order). Then for each Lánczos iteration d, $d \geq K$:

1: compute $\lambda_K^{(d)}$ starting bisection in $[\lambda_K^{(d-1)},\ 2\lambda_K^{(d-1)} - \lambda_K^{(d-2)}]$// if $\lambda_K^{(d-1)}$ or $\lambda_K^{(d-2)}$ are not available use Geršgorin theorem.
2: **if** $\lambda_K^{(d)} - \lambda_K^{(d-1)}$ is above desired precision **then**
3: do another iteration
4: **else**
5: halt
6: **end if**

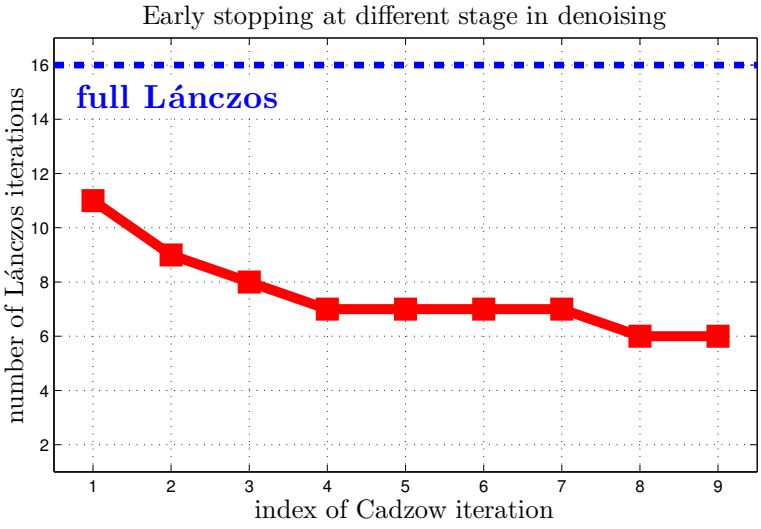


FIGURE 2.4: Early stopping with machine precision set to 10^{-8}.

Figure 2.4 shows an important number of iterations can be saved especially if a lot of eigenvalues are close to 0.

2.4 Partial eigenvalue decomposition of real, symetric tridiagonal matrices

2.4.1 Computation of the eigenvalues

The Lánczos algorithm results in $\text{proj}_{\mathcal{K}^m}\boldsymbol{A} = \boldsymbol{QTQ}^*$. The structure of $\boldsymbol{T}$ makes it easy to compute a particular subset of its eigenvalues. First the characteristic polynomial $p_m(\lambda)$ of $\boldsymbol{T}$ verifies:

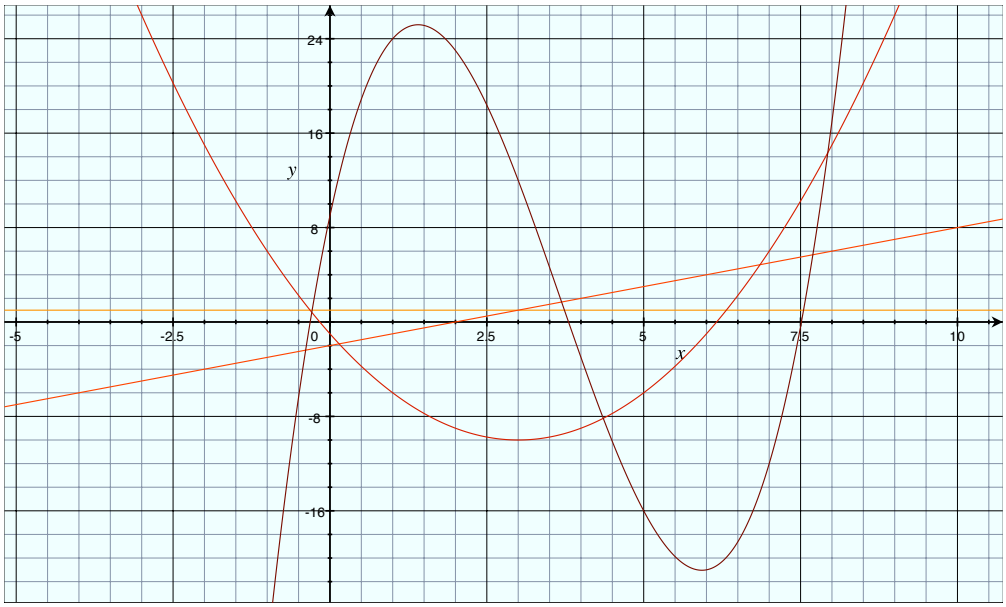
$$\begin{aligned} p_0(\lambda) = 1\ , \qquad p_1(\lambda) = \lambda - \alpha_1 \\ p_k(\lambda) = (\lambda - \alpha_k)p_{k-1}(\lambda) - \beta_k^2 p_{k-2}(\lambda)\ , \quad \forall k : 1 < k \leq m. \end{aligned} \tag{2.38}$$

To obtain the above recursion, simply expand $\det(\boldsymbol{T} - \lambda\mathbb{I})$. It provides the characteristic polynomials p_k of the sub-matrices $\boldsymbol{T}_{1:k}$ for $0 < k \leq m$ as well. Recalling the Sturm sequence property established in 2.12, the number of sign change(s) in the sequence $\{p_0(\lambda),\ \ldots,\ p_m(\lambda)\}$ is the number of eigenvalues greater than λ. Figure 2.5 provides an illustrative example on a small 3×3 matrix.

The ability to count eigenvalues makes it possible to find any k^{th} eigenvalue of $\boldsymbol{T}$ using a search algorithm like a bisection for example. Such an algorithm was described in [26] and is listed in appendix **??**. The bisection search has the nice property to minimize the maximum number of step to find a root over the set of continuous functions. However, the characteristic function is relatively smooth – it is a polynomial of finite degree m – which renders the bisection wasteful. Dekker & Brent[27] developped a root finding algorithm which by a clever choice of bisection, linear interpolation and quadratic inverse interpolation speeds up convergence for smooth functions while preserving the min-max property of the bisection. It can thus be combined with the Sturm sequence to yield the following algorithm:

1: isolate desired roots by bisection
2: use Brent algorithm on each interval containing exactly one root

2.4.1.1 Step 1: isolation of the eigenvalues



$$\boldsymbol{T} = \begin{pmatrix} 2 & 3 & 0 \\ 3 & 4 & 1 \\ 0 & 1 & 5 \end{pmatrix}$$

$$\begin{aligned} p_0(x) &= 1 \\ p_1(x) &= x-2 \\ p_2(x) &= (x-2)(x-4)-3^2 \\ p_3(x) &= (x-5)((x-2)(x-4)-3^2)-2^2(x-1). \end{aligned}$$

FIGURE 2.5: $\pm$ alternance of characteristic polynomials sequence counts smaller eigenvalues.

Algorithm 6 Finds intervals with exactly one eigenvalue.

Input: $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ 1st & 2nd diag. of $\boldsymbol{T}$; $0 < K \leq m$
Output: $\{(\inf_k, \sup_k)\}_{k=(m-K):(m-1)}$

```
m ← length(α)
x_min ← min_{i=1:K} |α_i| − |β_{i+1}| − |β_i|
x_max ← min(1, max_{i=1:m} |α_i| + |β_{i+1}| + |β_i|)
β ← β · β
[th_k^inf th_k^sup] ← [x_min x_max], k = (m − K) : (m − 1)
a ← 0
for k = (m − K) : (m − 1) do
  while a < k do
    th ← (th_k^inf + th_k^sup)/2
    a ← sturmCount(α, β, th)
    for i = (m − K) : (m − 1) do
      if q ≤ i then
        if th > th_i^inf then
          th_i^inf ← th
        end if
      else if th < th_i^sup then
        th_i^sup ← th
      end if
    end for
  end while
end for
return {(th_k^inf, th_k^sup)}_{k=(m−K):(m−1)}
```

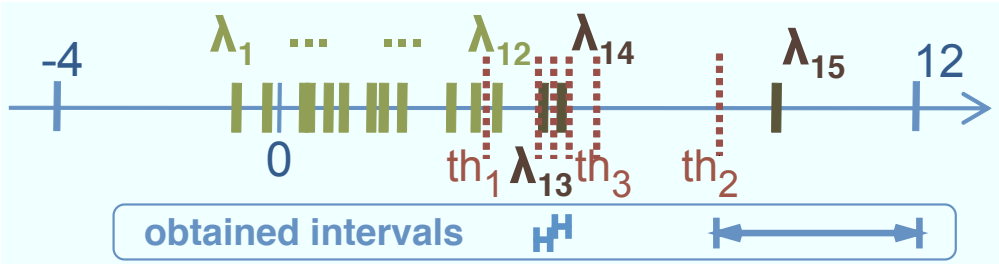
A method to find the K largest eigenvalues of $\boldsymbol{T}$ is listed in pseudo-code in algorithm 6. The behavior of the algorithm can be described in a few words:

- start with an interval $[x_{\min},\ x_{\max}]$ containing the eigenvalues $\lambda^+_{m-K},\ \ldots,\ \lambda^+_{m-1}$.
- to each of these eigenvalues λ_i^+ assign the pairs $(th_i^{inf}, q_i^{inf}) \stackrel{def}{=} (x_{\min}, m)$ and $(th_i^{sup}, q_i^{sup}) \stackrel{def}{=} (x_{\max}, 0)$. Each pair has the form (threshold th, # of eigenvalues $>$ th). Note the guess for $q^{\min}$ may be wrong.
- consider the eigenvalue with lowest index. Do a bisection search on the threshold until th^{inf} matches the index.
- at each step of the bisection with threshold th compute q the number of eigenvalues larger than th.
- update the pairs of current eigenvalue and the larger ones

  if q is smaller or equal to the eigenvalue index update the 'inf' pair if new threshold is tighter, else update the sup pair (if tighter).
- if th^{inf} matches the index of current eigenvalue, set next eigenvalue as current. Continue bisection.
- terminate when no eigenvalue left.

$(th,\ q)$	$\lambda_{13} = 5.1$	$\lambda_{14} = 5.3$	$\lambda_{15} = 9$
–	**(-4, 0) (12, 16)**	**(-4, 0) (12, 16)**	**(-4, 0) (12, 16)**
(4, 12)	**(4, 12)** (12, 16)	**(4, 12)** (12 16)	**(4, 12)** (12, 16)
(8, 15)	(4, 12) **(8, 15)**	(4, 12) **(8, 15)**	**(8, 15)** (12, 16)
(6, 15)	(4, 12) **(6, 15)**	(4, 12) **(6, 15)**	(8, 15) (12, 16)
(5, 13)	**(5, 13)** (6, 15)	**(5, 13)** (6, 15)	(8, 15) (12, 16)
(5.5, 15)	(5, 13) **(5.5, 15)**	(5, 13) **(5.5, 15)**	(8, 15) (12, 16)
(5.25, 14)	(5, 13) **(5.25, 14)**	**(5.25, 14)** (5.5, 15)	(8, 15) (12, 16)

(a) Updated values in **bold** and current eigenvalue in gray.



(b) spectrum and thresholds

FIGURE 2.6: Typical execution of algorithm 6 starting in interval $[-4,\ 12]$ to isolate the largest 3 eigenvalues.

Figure 2.6, shows the execution on a realistic (non-scaled) matrix.

There is one question left to answer: how to select an admissible initial interval. It is answered by *Geršgorin disc theorem*:

Theorem 2.13 (Geršgorin)**.**
Let $\boldsymbol{B}$ *be an* $m \times m$ *matrix such that* $\boldsymbol{B} = \boldsymbol{D} + \boldsymbol{F}$ *with* $\boldsymbol{D} = diag(d_1, \ldots, d_m)$ *and* F *diagonal entries are* 0*. Then:*

$$\lambda(\boldsymbol{B}) \subseteq \bigcup_{i=1}^{m} [d_i - r_i,\ d_i + r_i]$$

such that $r_i = \sum_{j=1}^{m} |f_{ij}|$ *for* $i = 1, \ldots, m$*.*

Proof. See [16], §8.1.2 □

In the symmetric tridiagonal case, it easily follows:

Corollary 2.14.
The K *largest eigenvalues of a real tridiagonal symmetric matrix are in* $[\lambda_{\min},\ \lambda_{\max}]$ *such that*

$$\begin{aligned} \lambda_{\min} &= \min_{i \in \{1,\ldots,K\}} \left(|\alpha_i| - |\beta_{i+1}| - |\beta_i|\right) \\ \lambda_{\max} &= \max_{i \in \{1,\ldots,m\}} \left(|\alpha_i| + |\beta_{i+1}| + |\beta_i|\right) \\ & \qquad \beta_1 = \beta_{m+1} \stackrel{def}{=} 0. \end{aligned}$$

Proof. Developping Geršgorin theorem for a tridiagonal symmetric matrix yield the identity on $\lambda_{\max}$ and $\tilde{\lambda}_{\min} = \min_{i \in \{1,\ldots,m\}} \left(|\alpha_i| - |\beta_{i+1}| - |\beta_i|\right)$. Recursive application of the Sturm sequence property yields: $\lambda_K(\boldsymbol{T}_K) < \lambda_K(\boldsymbol{T}_m)$. Applying Geršgorin theorem on the submatrix $\boldsymbol{T}_K$ completes the proof. □

2.4.1.2 Step 2: computation of an eigenvalue

We use Brent & Dekker root finding algorithm. As a reference, this particular algorithm – with some additional bells and whistles – is known as `fzero` in the MATLAB environement. A thorough explanation of the algorithm is found in [27]. A fixed-point ready code is listed in appendix A.1 as `fzeroS`. MATLAB code is listed in appendix **??** under the same name.

2.4.2 Computation of the eigenvector

The task looks trivial at first. Given an eigenvalue λ solve the singular homogeneous system $(\boldsymbol{T} - \lambda\mathbb{I})\boldsymbol{\xi} = \mathbf{0}$. Well *Algebra 101* tells us to fix any $\xi_i = 1$ and to remove the corresponding equation from the system – any i can be picked as $\boldsymbol{T}$ structure guarantees no single row of $\boldsymbol{T} - \lambda\mathbb{I}$ is independent to every other row. So let's remove the first equation. It amounts to get rid of the constraint $p_m(\lambda) = 0$, which after all has to be true since λ is an eigenvalue. The tridiagonal symmetric nature of $\boldsymbol{T}$ yields the following formula to compute $\boldsymbol{\xi}$:

$$\xi_1 = 1\ , \quad \text{(arbitarily set } \neq 0). \tag{2.39}$$

$$\xi_i = \xi_1 \frac{p_{i-1}(\lambda)}{\prod_{j=1}^{i-1} \alpha_j}\ , \quad j = 2,\ \ldots,\ m. \tag{2.40}$$

Now, try this new toy on a typical symmetric tridiagonal matrix, with an eigenvalue λ computed up to precision $10^{-8}/|\lambda| \approx 10^{-8}$. A true eigenvector was comupted using MATLAB `eig` function. Normalization was then applied to our estimated eigenvector by matching the 1^{st} coefficient. Here is the result:

estimated eigenvct.	true eigenvct.
-0.046318507639632	*-0.046318507639632*
-0.14068357<u>6369929</u>	-0.140683573722696
-0.06320945<u>5229830</u>	-0.063209448271466
0.09518955<u>2760338</u>	0.095189567740435
0.38215673<u>8749264</u>	0.382156771708248
0.74781668<u>0474655</u>	0.747816721172518
0.49388856<u>6871232</u>	0.493888553311323
0.12282874<u>4421528</u>	0.122828566005606
0.02604080<u>8557118</u>	0.026039945766723
0.00635366<u>6983591</u>	0.006350352290455
0.00119931<u>1810890</u>	0.001181334248749
0.00029447<u>6245805</u>	0.000178620974257
0.00136306<u>2825963</u>	0.000014911687636
0.01350192<u>9330903</u>	0.000001565397746
0.21381281<u>5573857</u>	0.000000088985210
<u>7.898210524988566</u>	0.000000002301723

Non-matching digits are <u>underlined</u>. First row was used for normalization.

One would expect accuracy of the eigenvector to match the one of the eigenvalue approximation. It is indeed true at the beginning of the recursion, but it soon gets out of hand: last entries should approach 0, however they grow above unit. Thus, the subspace spanned by the estimated eigenvector would bear little resemblance to the true subspace. The intuitive explanation is that instead of being a solution to the homogeneous system, it solves [28]:

$$(\boldsymbol{T} - \lambda\mathbb{I})\boldsymbol{\xi} = \mu_k \boldsymbol{e}_k \ , \quad \xi_k = 1. \tag{2.41}$$

with $\boldsymbol{e}_k$ the k^{th} vector of the canonical basis. $\boldsymbol{\xi}$ is in fact an eigenvector of the matrix $\boldsymbol{T}$ with the perturbation μ_k on the k^{th} diagonal entry. In finite-precision arithmetic

> "all equations are equal, but some are more equal than others"[2]

, *i.e.* the most redundant equation r shall be removed:

$$r = \arg\min_{k=1:m} \mu_k \tag{2.42}$$

There is no good index r known a-priori. This problem was well-known since the beginning of last century, and received a solution, first by Godunov [29], and then – independently – by Fernando [28]. We will focus on Fernando's solution as it is elegant and lends itself easily to an efficient implementation.[3]

2.4.2.1 Fernando's double factorization

It is noteworthy the normalized Sturm sequence formula implicitely describes an $\boldsymbol{L}\boldsymbol{D}_l\boldsymbol{L}^*$ factorization of $\boldsymbol{T} - \lambda\mathbb{I}$. Indeed:

$$\bar{p}_{i+1}(\lambda) = \frac{p_{i+1}(\lambda)}{p_i(\lambda)} = \alpha_i - \lambda - \frac{\beta_i^2}{\bar{p}_i}. \tag{2.43}$$

[2]However, we won't call them *Snowball* or *Napoleon*.

[3]Another reason is that the only free reference (not a book) I could find for Godunov's method contains errors...

yields the matrix factorization:

$$\boldsymbol{T} - \lambda\mathbb{I} = \boldsymbol{L}\boldsymbol{D}_l\boldsymbol{L}^*.$$

$$\boldsymbol{D}_l = \begin{pmatrix} \bar{p}_1 & & & \\ & \ddots & \mathbf{0} & \\ & & \bar{p}_k & \\ & \mathbf{0} & & \ddots \\ & & & \bar{p}_m \end{pmatrix}, \qquad L = \begin{pmatrix} 1 & & & & \\ \frac{\beta_2}{\bar{p}_1} & 1 & & & \mathbf{0} \\ & \ddots & \ddots & & \\ & & \frac{\beta_{k+1}}{\bar{p}_k} & 1 & \\ & \mathbf{0} & & \ddots & \ddots \\ & & & & \frac{\beta_m}{\bar{p}_{m-1}} \quad 1 \end{pmatrix}.$$

Similarly, starting the recursion from the other end yields the $\boldsymbol{U}\boldsymbol{D}_u\boldsymbol{U}^*$ factorization. In [28](eq.15) the following formula on the residual is proven:

$$\mu_k = D_l(k) + D_u(k) - (\alpha_k - \lambda). \tag{2.44}$$

The equation to be removed in the system minimizes equation 2.44. An efficient way to solve the system is to use the previously computed factorizations [28](thm.3). Starting with $\xi_r = 1$, the $\boldsymbol{L}\boldsymbol{D}_l\boldsymbol{L}^*$ factorization with lower diagonal $\boldsymbol{l} = [l_1, \ldots, l_{m-1}]$ provides the forward recursion:

$$\xi_i = -l_{j-1}\xi_{i-1}. \tag{2.45}$$

and the $\boldsymbol{U}\boldsymbol{D}_u\boldsymbol{U}^*$ factorization with upper diagonal $\boldsymbol{u} = [u_1, \ldots, u_{m-1}]$ provides the backward recursion:

$$\xi_{i-1} = -u_{i-1}\xi_i. \tag{2.46}$$

The eigenvectors obtained with this method have the same accuracy as the eigenvalue approximation they are computed from.

A MATLAB code listing is provided in appendix **??**.

2.5 Implementation & fixed-point arithmetic

The targeted hardware has basic $32 - bit$ arithmetic, boolean and comparative capabilities: $\pm$, $\times$, $\wedge$, $\vee$, $\neg$, $\oplus$, $\equiv$, $\neq$, $>$, ... In addition it has standard shift operations on registers – useful to multiply or divide by a power of 2 – and composite operation like *multiply accumulate* $\overset{\Sigma}{\times}$ (add result of a multiplication to a register), which is very useful for matrix multiplications or inner-products.

The signed fixed-point number format is written sQ$i.f$, where s stands for "signed", i is the number of integer bits, and f the number of fractional bits. The total number of bits is refered as $b = 1 + i + f$. The position of the fractional "." is called *radix*.

With this in mind, implementation issues boil down to two requirements:

1. guarantee accuracy: prevent or treat overflow/underflow suitably. If an algorithm satisfies this condition for every operation, it is called *stable*.
2. use cheap operations as often as possible

To enforce condition 1, we make the choice to downscale the data. Doing so allows to rule out overflow in many cases, leaving correct underflow handling the only remaining issue. Namely we assume $\frac{1}{2} \leq \|\boldsymbol{A}\|_F < 1$. It can be cheaply enforced as the matrix is Toeplitz and the interval allows division by a power of 2. Note scaling is not meaningful in itself as it is a simple shift of the radix. It however provides a reference, setting the matrix to have roughly unit energy.

Stability of the Lánczos iterations

Proposition 2.15.
The Lanczos algorithm 5 is stable for sQ$0.(b-1)$.

Proof. Since $\boldsymbol{Q}$ is unitary, $\|\boldsymbol{T}\|_F = \|\boldsymbol{A}\|_F$, thus $|\alpha_m|, |\beta_m| < \|\boldsymbol{A}\|_F$, $\forall m$. It follows $\|\boldsymbol{r}\| < \|\boldsymbol{Q}\|_F$ after both affectations since column vectors of $\boldsymbol{Q}$ have unit norm. For the same reasons, operations inside these two affectations yields result of norm less or equal to $\|\boldsymbol{A}\|_F$. □

The next step is to guarantee stability of the eigenvalue decomposition of $\boldsymbol{T}$.

Stability of the Sturm sequence computation

Proposition 2.16.
The (non-normalized) Sturm sequence $p_i(\lambda)$ verifies:

$$|p_m(\lambda)| < (1 + |\lambda|) \max\left(|p_{m-1}|, |p_{m-2}|\right)$$

for all $1 \leq i \leq M$

Proof. It is true for $i = 1$, thus $\exists p_i : \ |p_i(\lambda)| < (1 + |\lambda|) \max(|p_{i-1}(\lambda)|, |p_{i-2}(\lambda)|)$. then using the recursion formula and the identity $\alpha_i^2 + 2\beta_i^2 < 1$:

$$\begin{aligned} |p_i(\lambda)| &= \left|(\lambda - \alpha_i)p_{i-1}(\lambda) - \beta_i^2 p_{i-2}(\lambda)\right| \\ &< \left(|\lambda| + \sqrt{1 - 2\beta_i^2} + \beta_i^2\right) \max(|p_{i-1}(\lambda)|, |p_{i-2}(\lambda)|). \end{aligned}$$

A substitution $x \leftarrow \sqrt{1 - 2\beta_i^2}$ (remember $0 \leq \beta_i^2 < 1/2$) and derivation with respect to dx reveals strict growth in $]0\ 1[$, thus maximum is reached for $\beta_i^2 = 0$. □

The values of λ for which the Sturm sequence is going to be evaluated is bounded by

Proposition 2.17.
The interval $[x_{min}\ x_{max}]$ *is included in* $]-1\ \sqrt{2}[$

Proof.
The lower-bound holds trivially: $|x_{min}| < \max(|\alpha_i|, |\beta_i| + |\beta_{i-1}|)$ and $|\alpha_i| < 1$. Moreover $\beta_i^2 + \beta_{i-1}^2 < 1/2$. Since the objective function is monotonically increasing in $\beta_{i,i-1}$ optimum is reached for $\beta_i^2 = 1/2 - \beta_{i-1}^2$, i.e. $\beta_i^2 = \beta_{i-1}^2 = \frac{1}{4}$ (a symmetric argument would have done the job as well).

For the upper-bound, we follow the same reasonning, we can thus eliminate the variable α_i, and solve (we call the "β^2s" y and z for short):

$$\max_{y,z} f(y, z) = \sqrt{1 - 2(y + z)} + \sqrt{y} + \sqrt{z}, \quad \text{s.c. } 0 \leq y + z \leq 1/2.$$

The symmetry of the problem implies: $\boldsymbol{\nabla} f = \mathbf{0}$ iif $y = z$.Computing the partial derivative in y yields an extremum (which is a maximum) $4y_o = 1 - 4y_o$. Thus $\beta_i^2 = \beta_{i-1}^2 = \frac{1}{8} \Rightarrow \alpha_i^2 = \frac{1}{2}$. It follows $|x_{max}| < \sqrt{2}$. □

As an example, the upper-bound is reached by the following matrix (up to 0-padding):

$$\boldsymbol{T}_{max} = \frac{1}{2\sqrt{2}} \begin{pmatrix} 0 & 1 & 0 \\ 1 & 2 - \epsilon & 1 \\ 0 & 1 & 0 \end{pmatrix}, \text{ for } \epsilon \text{ small enough .}$$

It is then natural to use some extra knowledge to make the Geršgorin interval tighter setting $x_{max} = \min(1, x_{max})$.

Proposition 2.18.
The Sturm sequence computation is stable enforcing $p_{i-2}(\lambda)$ *has 2 leading zeros at each step* i.

Proof. Use proposition 2.16 with property $|\lambda| \leq 1$. □

Previous result draws an implementation of the non-normalized Sturm sequence avoiding overflow and making underflow as unlikely as possible. It mimics floating-point representation, in a very limited way.

Algorithm 7 Evaluation in]-1 1[of the characteristic polynomial of tridiagonal real symmetric matrix $\boldsymbol{T}$ s.t. $\|\boldsymbol{T}\|_F < 1$. Signed fixed pt. $Q2.(b-3)$ arithmetic.

Input: $\boldsymbol{\alpha}$, $\boldsymbol{\beta}^2$ 1st & and square of 2nd diag. of $\boldsymbol{T}$ s.t. $\|\boldsymbol{T}\|_F < 1$; λ a scalar s.t. $|\lambda| < 1$.
Output: $[f_l\ e_l] = p_M(f_l 2^{-e_l})$
  $e_l \leftarrow 0$; $\boldsymbol{\beta}^2 \leftarrow [1\ \boldsymbol{\beta}^2]$
  $p_2 \leftarrow 1$; $p_1 = \lambda - \alpha_1$
  for m=2:M **do**
    $s \leftarrow$ `leftmostbit`(p_2); $e_l \leftarrow e_l + s$
    $[p_1 p_2] \leftarrow$ `shiftleft`$([p_1 p_2], s)$
    $p \leftarrow (\lambda - \alpha_m)p_1 - \beta_m^2 p_2$
    $p_2 \leftarrow p_1$; $p_1 \leftarrow p$
  end for
  return $[p\ e_l]$

The function `leftmostbit` returns the position of the leftmost non-0 bit relative to the first fractional digit, *i.e.* `leftmostbit( 00.0101...011)=1 and leftmostbit( 10.0101...011)=-2`. It assumes big-endian format. Evaluation of the non-normalized characteristic polynomial is only required in algorithm 9, which is implemented to handle this particular $(fraction,\ exponent)$ format.

Stability of the `isolation` algorithm

For the isolation of eigenvalues and the $\boldsymbol{LD}_l\boldsymbol{L}$ factorization the normalized Sturm sequence is used. It is hard to determine if such a sequence can be stabily computed with an agressive fixed-point format, only a weak guarantee holds:

Proposition 2.19.
Provided a stable implementation of `sturmCount`*,* `isolation` *is stable for* $Q1.(b-2)$.

Proof. $\boldsymbol{\beta} \cdot \boldsymbol{\beta}$ is stable as $|\beta_i| < \|\boldsymbol{A}\|_F \leq 1 \Rightarrow \beta_i^2 < |\beta_i|$. Next notice the objective function and constraint are symmetric, thus maximum is reached for $|\alpha_i| = |\beta_i| = |\beta_{i+1}| = 1/\sqrt{3} < 2/3$. □

Proposition 2.20.
The (normalized) Sturm sequence $q_i(\lambda)$ *verifies:*

$$|q_i(\lambda)| < (1 + |\lambda|) \max\left(1, \frac{1}{|q_{i-1}|}\right).$$

for all $1 \leq i \leq M$ *and* $|\lambda| < 1$

Proof. The upper-bound holds trivially from $p_i(\lambda) < (1 + |\lambda|) \max(|p_{i-1}(\lambda)|, |p_{i-2}(\lambda)|)$ and the definition $q_i(\lambda) = \frac{p_i(\lambda)}{p_{i-1}(\lambda)}$. □

Proposition 2.21.
Using b-bit words and signed fixed point representation $sQ\left\lceil\frac{b-1}{2}\right\rceil.\left\lfloor\frac{b-1}{2}\right\rfloor$, `sturmCount` *is stable with the modification:*

$$q \leftarrow \alpha_i - \lambda - (q \neq 0)?\frac{\beta_i^2}{q} : 2^{\frac{b}{2}-1}|\beta_i|$$

Proof. The third term is upper-bounded by $2^{\frac{b}{2}-1}$, thus $q < 2^{\frac{b}{2}-1} + 2\|\boldsymbol{A}\|_F < 2^{\frac{b}{2}}$. Thus q cannot overflow. □

Proposition 2.20 yields the rather poor $\mathrm{sQ}\left\lceil\frac{b-1}{2}\right\rceil.\left\lfloor\frac{b-1}{2}\right\rfloor$. It is good enough to count $\pm$ alternance in the sequence (bisection), but the eigenvector computation will require an implementation with a moving radix point, which is not listed in this report.

2.6 Numerical results (under construction)

Simulation were performed with the code listed in **??**. Notice it uses `double` floating-point arithmetic, and is thus not a faithful representation of computations on LP-BAN . However, precision was limited to 10^{-8} in the computation of the eigenvalues and in the PRO routine. It is a little less than the achievable precision in sQ2.29. All steps but the computation of the eigenvectors and the PRO were shown to work for such a fixed point representation and input condition $\frac{1}{2} \leq \|A\|_F < 1$. The non-compliant code will need to be tweaked at implementation to avoid overflow.

Despite these limitations, this framework shall provide faithful information on the expected performances.

First accuracy of the method will be assessed in comparison to the classical full-precision `svd` based Cadzow denoising routine. Then a precise count of operations will be given based on the `ops` inserts in the MATLAB code. Finally the method will be compared to the state of the art, *i.e.* what would LAPACK do?

2.6.1 Accuracy measure

.

The stopping criterion in the classical Cadzow routine is $\epsilon > \frac{\sigma_{K+1}}{\|\boldsymbol{y}\|}$ if the target rank is K, σ_{K+1} is the smallest singular value of the $(M-K-1)\times(K+1)$ toeplitz matrix built from the vector of fourier coefficients $\boldsymbol{y}$. The classical value used to report results in previous chapters is $\epsilon = 10^{-10}$. Limiting precision to 10^{-8} in the Krylov subspace method yielded $\epsilon_{\mathcal{K}} \approx 5\times 10^{-7}$ in 9 Cadzow iterations.

2.6.2 Complexity

For a typical number of coefficients (31) and a target rank of 3, the entire denoising cost is estimated to:

```
      add: 100k
     mult: 100k
      div: 1.2k
     sqrt: 200
    shift: 10k
      clz: 5k
32pts-FFT: 300
     bool: 9k
```

2.6.3 Comparison with LAPACK

It is good to look at the state of the art, to avoid "homebrewing" suboptimal algorithms. Given a Toeplitz hermitian matrix, a full SVD will be done by:

- full Lánczos iterations
- QL/QR factorization of the tridiagonal matrix

A more economical version extracting only a few eigenpairs would be:

- full Lánczos iterations
- bisection search to compute the eigenvalues
- inverse iterations to compute the eigenvectors

The skeleton used in this report is similar to a numerically savvy use of LAPACK. However there are 2 important differences:

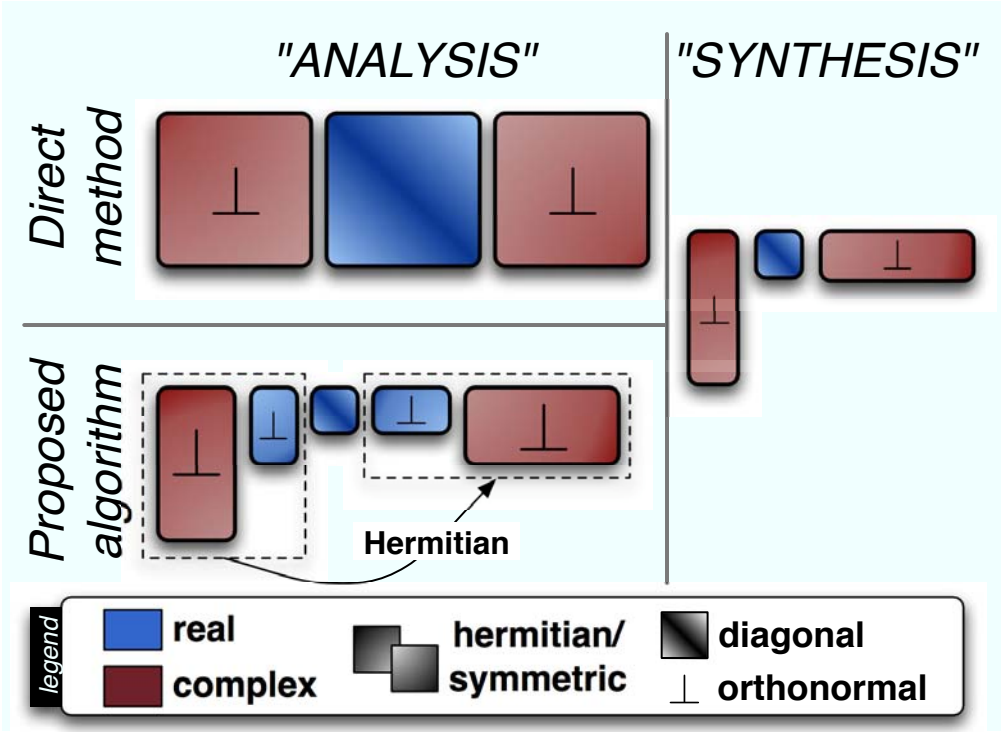
- only a few Lánczos iterations are done, especially after a few cadzow iterations when a lot of eigenvalues are close to 0.
- inverse iterations are avoided using a very cheap double factorization of $\boldsymbol{T}$

The comparison was made to the best of my knowledge: LAPACK does not seem to provide the double factorization nor the faculty to stop the Lánczos algorithm early.

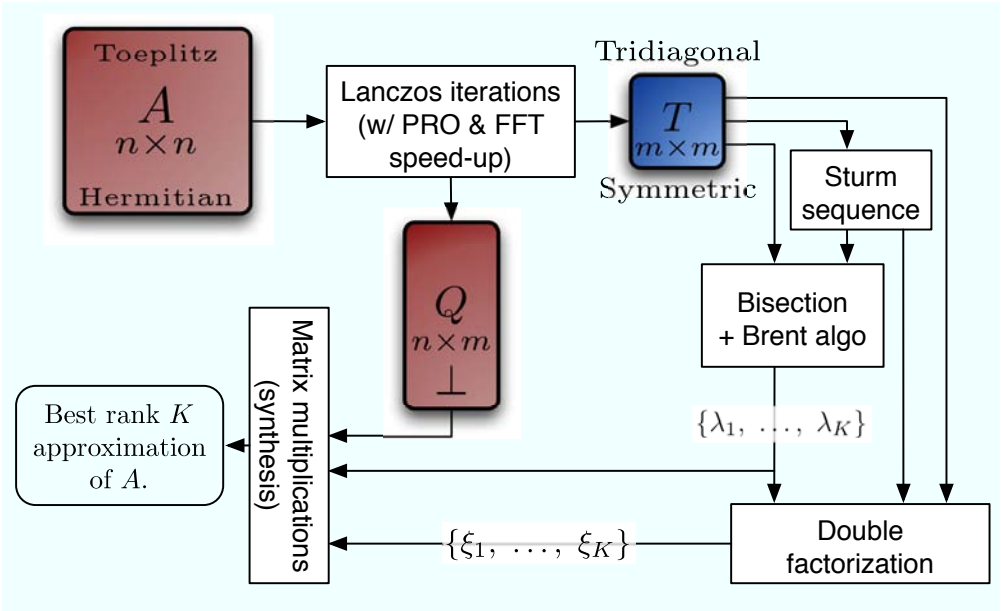
2.7 Chapter digest

The main point made in this chapter is a *best low-rank approximation* of an hermitian matrix can be found in a *Krylov subspace* of much smaller dimension. The availability of a relatively fast and efficient algorithm – the *Lánczos iterations* – to perform the projection makes this approach relevant. Moreover, connection between Krylov subspaces and polynomials of a certain finite degree revealed the projection has a *symmetric tridiagonal* structure and its eigenpairs can be selectively computed, giving the proposed method an additional edge over the traditional full-SVD approach. The computational gain is particularly clear in figure 2.7(a). As well, the relatively low count of divisions and square-root operations and some guarantees on fixed-point computations makes it suitable for an implementation on embedded devices with limited arithmetical capabilities.

A flow chart provided in figure 2.7(b) summarize the algorithm.



(a) Comparison of computed matrices in the full-SVD approach to the Krylov subspaces based algorithm.



(b) Flow-chart of the Krylov subspaces based algorithm. Dimensions compares to $K \ll m \ll n$.

FIGURE 2.7: Summary of the proposed algorithm properties.

Appendix A

Fast Cadzow denoising, code listing

A.1 Brent algorithm: pseudo code

The main difficulty is to compute accurately a ratio of characteristic polynomials. It is solved by a representation $(fraction,\ exponent)$. The ratio of two of these numbers is then computed via the `divfe` function. The function `charpol` is listed in algorithm 7

Algorithm 8 `divfe`: Division of scalars represented in in the [fraction exponent] format. Result must have magnitude less than 2^i. Signed fixed pt. $Qi.(b-i-1)$ arithmetic

Input: $f_n,\ f_d$ s.t. $|f_n|, |f_d| < 4$ and $e_n,\ e_d$ integers. Supposes $\left|\frac{f_n 2^{-e_n}}{f_d 2^{-e_d}}\right| < 2^i$.
Output: $s = \frac{f_n 2^{-e_n}}{f_d 2^{-e_d}}$
  if `leftmostbit`$(f_n) + e_n - e_d < -i$ **then**
    $s \leftarrow$ `shiftleft`$(\frac{f_n}{f_d}, e_n - e_d)$
  else
    $s \leftarrow \frac{\texttt{shiftleft}(f_n, e_n - e_d)}{f_d}$
  end if
  return s

Algorithm 9 `zeroS`: Find the unique zero of the characteristic polynomial in an interval (signed fixed-pt sQ2.$(b-3)$)

Input: $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ 1st & 2nd diag. of $\boldsymbol{T}$; $[a,\ b]$ an interval s.t. $b-a \leq \frac{1}{2}$, ε a maximum error ($\varepsilon \geq 2^{4-b}$)

Output: z an approximation of z_0 s.t. $p_M(z_0) = 0$ and $|z - z_0| \leq \varepsilon$.

$tol \leftarrow \frac{\varepsilon}{2}$; $[fa\ ea] \leftarrow \texttt{charpol}(\boldsymbol{\alpha}, \boldsymbol{\beta}^2, a)$; $[fb\ eb] \leftarrow \texttt{charpol}(\boldsymbol{\alpha}, \boldsymbol{\beta}^2, b)$; $[fc\ ec] \leftarrow [fb\ eb]$; $c \leftarrow a$

while $|c-b| > \varepsilon$ **do**

if $\neg(fb > 0 \oplus fc > 0)$ **then**

$c \leftarrow a$; $[fc\ ec] \leftarrow [fa\ ea]$; $d \leftarrow b-a$; $e \leftarrow d$;

end if

$clb \leftarrow (eb > ec)?(|\texttt{shiftleft}(fc, ec-eb)| < |fb|) : (|fc| < |\texttt{shiftleft}(fb, eb-ec)|)$

if clb **then**

$a \leftarrow b$; $b \leftarrow c$; $c \leftarrow a$; $[fa\ ea] \leftarrow [fb\ eb]$; $[fb\ eb] \leftarrow [fc\ ec]$; $[fc\ ec] \leftarrow [fa\ ea]$

end if

$m \leftarrow \frac{c-b}{2}$

$alb \leftarrow (eb > ea)?(|\texttt{shiftleft}(fa, ea-eb)| < |fb|) : (|fa| < |\texttt{shiftleft}(fb, eb-ea)|)$

if $(|e| < \varepsilon) \vee alb$ **then**

$d \leftarrow m$; $e \leftarrow m$

else

$s \leftarrow \texttt{divfe}(fb, eb, fa, ea)$;

if $a \equiv c$ **then**

$p \leftarrow 2 \times m \times s$; $q \leftarrow 1 - s$

else

$q \leftarrow \texttt{divfe}(fa, ea, fc, ec)$; $r \leftarrow q \times s$

$p \leftarrow s \times (2 \times m \times q \times (q-r) - (b-a) \times (r-1))$; $q \leftarrow (q-1) \times (r-1) \times (s-1)$

end if

$[p\ q] \leftarrow (p > 0)?[p\ \ -q] : [-p\ q]$

if $(2 \times p < 3 \times m \times q) \wedge (p < \frac{|s \times q|}{2})$ **then**

$e \leftarrow d$; $d \leftarrow \frac{p}{q}$

else

$d,\ e \leftarrow m$

end if

end if

$a \leftarrow b$; $[fa\ ea] \leftarrow [fb\ eb]$; $b \leftarrow b + (|d| > tol)?d : (m > 0)?tol : -tol$; $[fb\ eb] \leftarrow \texttt{charpol}(\boldsymbol{\alpha}, \boldsymbol{\beta}^2, b)$

end while

return $z \leftarrow b$

Bibliography

[1] T. Blu, P.-L. Dragotti, M. Vetterli, P. Marziliano, and L. Coulot. Sparse sampling of signal innovations. *IEEE Signal Processing Magazine*, 25(2):31–40, March 2008. URL `http://bigwww.epfl.ch/publications/blu0801.html`.

[2] P. Marziliano. *Sampling Innovations*. PhD thesis, Communication Systems Department, EPFL, Lausanne, Switzerland, April 2001. URL `http://www3.ntu.edu.sg/home/epina/Publications/PinaMarzilianoThesisBook.pdf`.

[3] Martin Vetterli, Pina Marziliano, and Thierry Blu. Sampling signals with finite rate of innovation. *IEEE Trans. Signal Process.*, 50(6):1417–1428, 2002. ISSN 1053-587X.

[4] 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*, 55(5):1741–1757, May 2007. URL `http://www.commsp.ee.ic.ac.uk/~pld/publications/DragottiVB_SP06.pdf`. Part 1.

[5] J.A. Cadzow. Signal enhancement-a composite property mapping algorithm. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 36(1):49–62, Jan 1988. ISSN 0096-3518. doi: 10.1109/29.1488.

[6] E. S. Pearson. Note on an approximation to the distribution of non-central χ^2. *Biometrika*, 46(3-4):364, 1959. doi: 10.1093/biomet/46.3-4.364. URL `http://biomet.oxfordjournals.org/cgi/reprint/46/3-4/352.pdf`.

[7] J. P. Imhof. Computing the distribution of quadratic forms in normal variables. *Biometrika*, 48(3-4):419–426, 1961. doi: 10.1093/biomet/48.3-4.419. URL `http://biomet.oxfordjournals.org/cgi/reprint/48/3-4/419.pdf`.

[8] H. Scheffé. *Analysis of variance*. John Wiley and Sons, London, 1959.

[9] H. Liu, Y. Tang, and H. H. Zhang. A new chi-square approximation to the distribution of non-negative definite quadratic forms in non-central

normal variables. *Computational Statistics & Data Analysis*, 53(4):853 – 856, 2009. ISSN 0167-9473. doi: 10.1016/j.csda.2008.11.025. URL http://www.sciencedirect.com/science/article/B6V8V-4V353XG-2/2/83450bca4ff6e29b95

[10] Ali Hormati, Olivier Roy, Yue M. Lu, and Martin Vetterli. Distributed Sampling of Correlated Signals Linked by Sparse Filtering: Theory and Applications. *IEEE Transactions on Signal Processing*, 2009.

[11] I. J. Schönberg. Contributions to the problem of approximation of equidistant data by analytic functions. *Quart. Appl. Math.*, 4:45Ű99 and 112Ű141, 1946.

[12] M. Unser. Splines: A perfect fit for signal and image processing. *IEEE Signal Processing Magazine*, 16(6):22–38, November 1999. URL http://bigwww.epfl.ch/publications/unser9902.html. IEEE Signal Processing Society's 2000 magazine award.

[13] H. Cramér. *Mathematical methods of statistics*. Princeton University Press, Princeton NJ, 1946.

[14] C. R. Rao. Information and accuracy attainable in the estimation of statistical parameters. *Bull. Calcutta Math. Soc.*, 37:81–91, 1945.

[15] B. Porat and B. Friedlander. Computation of the exact information matrix of gaussian time series with stationary random components. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 34(1):118–130, Feb 1986. ISSN 0096-3518.

[16] G. H. Golub and C. F. van Loan. *Matrix Computations*. Johns Hopkins University Press, 3rd edition, 1996.

[17] B. N. Parlett. *The Symmetric Eigenvalue Problem*. SIAM, Philadelphia,PA, 1998.

[18] R. Courant and D. Hilbert. *Methods of Mathematical Physics*, volume 1. Interscience Publishers Inc., New-York USA, 1953.

[19] T. S. Chihara. *An Introduction to Orthogonal Polynomials*. Gordon and Breach, New York, USA, 1978.

[20] C. Lánczos. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. *J. Res. Nat. Bur. Standards*, 45:255–282, 1950.

[21] G.C. Danielson and C. Lánczos. Some improvements in practical fourier analysis and their application to x-ray scattering from liquids. *Journal of the Franklin Institute*, 233(4 and 5):365–380 and 432–452, 1942.

[22] Milton Abramowitz and Irene A. Stegun. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. Dover, New York, ninth dover printing, tenth gpo printing edition, 1964. ISBN 0-486-61272-4.

[23] H. D. Simon. The lanczos algorithm with partial reorthogonalization. *Mathematics of Computation*, 42(165):115–142, January 1984.

[24] Gabor Szegő. *Orthogonal Polynomials*. Colloquium Publications - American Mathematical Society, 1939. ISBN 0-8218-1023-5.

[25] C. C. Paige. *The Computation of Eigenvalues and Eigenvectors of Very Large Sparse Matrices*. PhD thesis, London University, London, UK, 1971.

[26] W. Barth, R. Martin, and J. Wilkinson. Calculation of the eigenvalues of a symmetric tridiagonal matrix by the method of bisection. *Numerische Mathematik*, 9 (5):386–393, April 1967. URL `http://dx.doi.org/10.1007/BF02162154`.

[27] Richard P. Brent. *Algorithms for Minimisation Without Derivatives*. Prentice Hall, 1973. ISBN 0130223352.

[28] K. V. Fernando. Accurate babe factorisation of tridiagonal matrices for eigenproblems. Technical report, NAG Ltd, 1995.

[29] S.K. Godunov, A.G. Antonov, O.P. Kiriljuk, and V.I. Kostin. *Guaranteed Accuracy in Numerical Linear Algebra*. Kluwer Academic, Dordrecht, 1993.