Convex coders and oversampled A/D conversion: theory and algorithms

Nguyen T. Thao and Martin Vetterli
Department of Electrical Engineering
and Center for Telecommunications Research
Columbia University, New York, NY 10027-6699

December 16, 1991

\(^1\)Work supported in part by the National Science Foundation under grant ECD-88-11111.
Abstract

Signal reconstruction in oversampled A/D conversion (ADC) is classically performed by lowpass filtering the quantized signal. This leads to a mean squared error (MSE) inversely proportional to $R^{2n+1}$ where $R$ is the oversampling rate and $n$ is the order of the converter. However, while the estimate given by this reconstruction has the same bandwidth as that of the original analog signal, we show that it does not necessarily lead to the same digital sequence when fed into the same A/D converter. Moreover, under some assumptions, we show analytically that an estimate having the same bandwidth and giving the same digital sequence as those of the original signal should yield an MSE with an upper bound inversely proportional to $R^{2n+2}$, instead of $R^{2n+1}$; that is an improvement of 3 dB per octave of oversampling, regardless of the order of the converter.

We propose a structural analysis covering most currently known configurations of oversampled ADC, which enables us to identify sets of input analog signals giving the same digital sequences. This is based on the direct knowledge of the conversion mechanisms and includes circuit imperfections if they can be measured. The inherent property of convexity of these sets gives us theoretical means to achieve estimates which have the same bandwidth as and reproduce the digital sequence of a given input signal. We designed computer implementable algorithms which achieve such estimates. Using these algorithms, we performed numerical experiments on sinusoidal inputs which confirm the $R^{2n+2}$ behavior of the signal reconstruction MSE. Moreover, we do not observe any performance decay in keeping this $R^{2n+2}$ behavior, when converters include circuit imperfections up to 1%.
Contents

1 Introduction .................................................. 3

2 Analysis of ADC signal partition ..................................... 10
  2.1 Simple ADC .................................................. 10
  2.2 Single-quantizer conversion ..................................... 11
  2.3 Convexity of signal cells and projection .......................... 14
  2.4 Signal cell of a code subsequence, convexity and projection ....... 17
  2.5 Multi-quantizer conversion ..................................... 19
  2.6 Conclusion .................................................... 21

3 MSE analysis of optimal signal reconstruction in oversampled ADC ............ 23
  3.1 Modelization of bandlimited and periodic signals \( V_0 \) .................. 23
  3.2 Simple ADC .................................................. 24
  3.3 \( n^{th} \) order single-quantizer converter ........................... 26
  3.4 \( n^{th} \) order multi-quantizer converter ............................. 34
  3.5 Conclusion .................................................... 35

4 Algorithms for the projection on \( \Gamma_1(C_0) \) .................................. 36
  4.1 Introduction ................................................... 36
  4.2 Specific formulation of the constrained minimization problem and properties .................................................. 37
  4.3 Zero\(^{th}\) order converter (simple, dithered, predictive ADC) ............ 40
  4.4 \( 1^{st} \) order converter (\( 1^{st} \) order \( \Sigma \Delta \) modulation) .................. 40
  4.5 Higher order converter (\( n^{th} \) order \( \Sigma \Delta \) modulation) ............... 42
    4.5.1 Introduction .............................................. 42
    4.5.2 Recursion principle and first algorithm ........................... 43
    4.5.3 More practical version for algorithm 3.1 .......................... 45
    4.5.4 Computation of \( W_{p,i}, V_p^0 \) .................................. 48
    4.5.5 Recapitulation ............................................ 49
    4.5.6 Algorithms 3.4, 3.5 with limited buffer .......................... 50

5 Numerical experiments ........................................... 52
  5.1 Validation of the \( R^{-2(n+2)} \) behavior of the MSE with sinusoidal inputs .... 52
  5.2 Efficiency of projection algorithms for \( n \geq 2 \) ......................... 53
  5.3 Specific tests for \( 1^{st} \) and \( 2^{nd} \) order \( \Sigma \Delta \) modulation ........... 54
  5.4 Further improvements with relaxation coefficients ...................... 55
  5.5 Circuit imperfections ......................................... 55
# Appendix

A.1 Proof lemmas 3.2 and 3.5 ........................................... 59
A.1.1 Preliminary notations and lemmas .............................. 59
A.1.2 Proof of lemma 3.2 ........................................... 60
A.1.3 Proof of lemma 3.5 ........................................... 62
A.1.4 Proof of lemma A.2 ........................................... 62
A.1.5 Proof of lemma A.5 ........................................... 63
A.2 Proof of fact 4.5 .................................................. 64
A.3 Proof of fact 4.7 .................................................. 64
A.4 Proof of fact 4.12 .................................................. 66
A.4.1 Preliminaries .................................................. 66
A.4.2 Proof of fact 4.12 ........................................... 67
A.4.3 Analytical expressions of $M(l)$ and $N(l)$ versus $l$ .... 68
A.4.4 Proof of fact A.11 ........................................... 69
A.4.5 Proof of fact A.12 ........................................... 70
A.5 Proof of fact 4.14 .................................................. 71
A.6 Study of algorithm 3.4 in the case $n = 2$ ....................... 73
A.6.1 Expression of $W_{p,i}^{0}, V_{p}^{0}$ .............................. 73
A.6.2 Proof of conjecture 4.6 in the case $n = 2$ .................... 73
A.6.3 Exponential decay of $W_{p,i}^{0}$ for increasing $p$ .............. 73
A.7 Proof of fact 4.15 .................................................. 74
Chapter 1

Introduction

Analog to digital conversion (ADC) is classically viewed as an approximation of an analog signal by another signal which has a discrete representation in time and amplitude. When the sampling rate is greater than or equal to the Nyquist rate, no information is lost after discretization in time. The loss of accuracy comes only from amplitude quantization (discretization in amplitude). The basic motivation in high resolution ADC is to minimize the error signal, when reconstructing the analog signal from the quantized signal. This minimization is often based on statistical considerations. For example, when the ADC consists of a simple uniform quantizer, it is assumed that the error signal has a uniform distribution and a flat spectral density. These assumptions have been the starting point for the development of oversampled ADC. While this model of quantization error is not completely justified [10], it leads to good predictions of practical results. In the case of simple ADC (simple quantization of the sampled signal), the error signal power is expected to be reduced in proportion to the oversampling rate \( R \) by a lowpass filtering of the quantized signal \( R = \frac{f_s}{f_m} \), where \( f_s \) is the sampling frequency and \( f_m \) the maximum frequency of the input signal. The white noise assumption has also been the foundation for the design of noise shaping converters (multi-loop, multi-stage \( \Sigma \Delta \) or interpolative modulators) [1, 4, 6, 8]. When the noise shaping is performed by purely integrating filters, the power of the error signal contained in the digital output can be reduced in proportion to \( R^{2n+1} \) by a single lowpass filtering [2]. These ADC techniques turned out to be very successful due to their simplicity of implementation and the good performances obtained in signal reconstruction with real time linear processing [5].

However, in order for the assumption of white quantization noise to be verified in practice, some constraints have to be imposed on the ADC operation. For example, in simple ADC, the quantization noise looses its ‘whiteness’ when the quantizer resolution is too coarse. In every type of ADC, a certain level of linearity and matching has to be achieved by the in-built analog circuits of the converter (quantization thresholds, D/A feedbacks). The sensitivity to circuit imperfections increases with the complexity of the converter and becomes a limitation to high ADC resolution.

While the white quantization noise assumption has been the driving force in the development of most oversampled A/D converters, our motivation is to study \textbf{optimal signal reconstruction} without any statistical criteria and to study it directly from the knowledge of the converter’s mechanisms. In our sense, optimality is achieved when the reconstructed signal meets every characteristic known from the original signal. In particular, we will require the estimate to give the same digital output sequence as that of the original signal when fed into the same converter.
The motivation behind this criterion originally comes from the simple situation shown in figure 1.2. We plot an example of bandlimited analog signal generated numerically and oversampled by 4. The crosses show the result of classical reconstruction in oversampled ADC, obtained by lowpass filtering the quantized signal (black dots). The figure shows that this estimate does not yield the same digital sequence as the original. This can be seen at time indices 10 and 11. Let us now perform the following transformation: we project the two critical samples of the estimate \((k = 10 \text{ and } k = 11)\) to the nearest border of the quantization interval which actually contains the original samples. This interval can be identified from the knowledge of the original digital sequence. We end up with a new estimate which yields the same digital sequence as that of the original analog signal and is at the same time better than the first estimate: the distance between the first estimate and the original signal has been necessarily reduced. This improvement is automatic and does not depend on any assumption about the variations of the analog signal.

Further remarks are appropriate to this simple example. Our estimate transformation does not require that quantization be uniform. It is based on the knowledge of the thresholds' position and is applicable whether the nonuniformity was designed or comes from circuit imperfections. In the latter case we simply require the quantization thresholds of the converter to be measurable. As a second remark, we do not even require the thresholds to be constant in time, provided that their variations are known or can be determined. This is typically the situation of dithered ADC. The particular condition here is that the dither be known in the time domain. This is also the less obvious situation of \(\Sigma \Delta\) and interpolative modulations, where, as will be shown in chapter 2, the feedback loop can be seen as a computable input dither. Our last remark is the following: the reconstruction improvement of figure 1.2 can be pushed further. Indeed, the new estimate has no reason to be bandlimited like the original signal. Therefore, it can be further improved with a second lowpass filtering. Again, the signal thus obtained may not yield the same digital output: the improvement procedure can be reiterated.

This simple example is in fact the starting point to our approach to ADC. What was precisely used in this example was the knowledge provided by the digital output of certain constraints in amplitude that the input signal should meet. Qualitatively speaking, once these constraints are identified, we can improve any estimate which does not meet them by a projection. If the digital sequence given by the input signal is \(C\), these constraints can be expressed as the set of all possible signals giving the same code sequence \(C\). Conceptually, the description of an A/D converter can be reduced to a many-to-one mapping between analog signals and digital sequences. The definition of the mapping only depends on the knowledge of the conversion's mechanisms (imperfections included). If this mapping can be determined, then the set of signals giving the same digital sequence \(C\) is the inverse image of \(C\) through the mapping. We denote this set by \(\Gamma(C)\) and call it the signal cell of \(C\). Since every analog signal necessarily gives one and only one digital sequence, the inverse mapping associated with the converter defines a partition of the space of input signals \(V\) (figure 1.3). We call it the signal partition associated with the A/D converter.

To identify estimates which have the same digital output as that of an original analog signal, it is therefore essential to determine the signal partition defined by the A/D converter. In chapter 2, we show how to derive the mapping associated with a converter, and thus the signal partition, on most A/D converters currently known: simple, dithered, predictive and noise shaping ADC, including multi-stage modulators. We will observe that these converters naturally yield signal partitions where every cell is convex. This property justifies the strong fact that, if an estimate \(X\) of an analog signal \(X_0\) giving the digital sequence \(C_0\) does not
Figure 1.1: Principle of simple oversampled A/D conversion and classical reconstruction.

Figure 1.2: Numerical example of bandlimited signal oversampled by 4. The signal $X^{(1)}$ obtained by lowpass filtering of the quantized signal $X^{(0)}$ does not give the same digital sequence as that of $X_0$. 
Figure 1.3: Signal space partition defined by an A/D converter: \( \Gamma(C_i) \) is the subset of all possible signals giving the digital sequence \( C_i \).

belong to \( \Gamma(C_0) \), the projection \( X' \) of \( X \) on \( \Gamma(C_0) \) is necessarily closer to \( X_0 \), wherever \( X_0 \) is located inside \( \Gamma(C_0) \) (figure 1.4). The signal transformation shown in figure 1.2 is just the realization of this projection in the particular case of simple ADC.

Now comes the obvious question: is this improvement to signal reconstruction substantial? The answer is of course no, when there is completely independence between the samples, like in Nyquist rate sampling. But when considering the oversampling situation, the samples of an input signal contain a certain redundancy. In other words, another restriction is known about the sampled input signals: they are confined to belong to the subspace \( V_0 \) of discrete time signals with the limited bandwidth related to the oversampling rate. This defines actually another convex specification about the sources signals. We will thus look for estimates which also meet this constraint. In other words, when \( X_0 \in V_0 \) gives the digital code \( C_0 \), we will try to find an estimate in the more restricted convex set \( V_0 \cap \Gamma(C_0) \).

Figure 1.4: Projection of \( X \) on the signal cell \( \Gamma(C_0) \) containing \( X_0 \). When \( \Gamma(C_0) \) is convex, this leads to an estimate \( X' \) which is necessarily closer to \( X_0 \).
In this context, the converter can be thought as defining a partition in the restricted space $V_0$ (figure 1.5).

This point of view gives an alternative interpretation of the work of Hein and Zakhor on $\Sigma\Delta$ modulators with constant inputs [9, 11, 12]. In that case, $V_0$ is a one dimensional space (amplitude of the constant), the A/D converter defines a partition of the real line and $V_0 \cap \Gamma(C_0)$ is an interval. In [9, 11], estimates for the MSE between $X_0$ and the center of the interval were given for 1st and 2nd order $\Sigma\Delta$ modulation.

In chapter 3, we use our point of view for MSE bounds in the more general case where signals in $V_0$ are bandlimited and periodic for simple ADC and any order multi-loop, multi-stage $\Sigma\Delta$ modulation, regardless of how the estimate $X$ is chosen inside $V_0 \cap \Gamma(C_0)$. With simple ADC, we prove that, if an input signal has at least a certain number of quantization threshold crossings within one period, the MSE at least decreases with the oversampling rate $R$ proportionally to $R^2$, instead of $R$ in classical reconstruction. This represents an improvement of 3 dB per octave of oversampling. The analysis of MSE is more difficult for higher order converters such as $n^{th}$ order $\Sigma\Delta$ modulators. However, if we start from the assumption of white quantization noise which leads to the noise reduction order of $R^{2n+1}$ in classical signal reconstruction, we prove that, taking an estimate of $X_0$ in $V_0 \cap \Gamma(C_0)$ yields a noise reduction of the order of $R^{2n+2}$ [14]. This, again, represents an improvement of 3 dB per octave of oversampling, regardless of the converter’s order. We then explain qualitatively how correlated quantization error can be considered in the derivation of $R^{2n+2}$.

These results give us strong motivation in finding computational means for such signal reconstruction. We know from Youla’s theorem [3] that alternating the projection of a first estimate on two convex sets necessarily converges to an element of the intersection (figure 1.6). While the projection on $V_0$ is an ideal lowpass filter with the specified bandwidth of input signals, we study in chapter 4 computer implementable algorithms for the projection on $\Gamma(C_0)$. Performing a projection on a convex set is a classic problem of constrained quadratic minimization. But in our case, we have the extra difficulty that a computer can only see and
process a signal locally in time. Taking this time constraint into account, we manage to find algorithms which perform the exact projection on $\Gamma(C_0)$ for zero$^{th}$ and 1$^{st}$ order converters. This includes simple, dithered, predictive ADC and 1$^{st}$ order $\Sigma\Delta$ modulation. Dealing with higher order conversion such as $n^{th}$ order multi-loop, multi-stage $\Sigma\Delta$ modulation, we derive algorithms which perform the projection on a larger convex set than $\Gamma(C_0)$ but which always contains $\Gamma(C_0)$ and therefore the solution $X_0$.

In chapter 5, we test these algorithms numerically on sinusoidal signals for simple ADC and 1$^{st}$ to 4$^{th}$ order multi-loop, multi-stage $\Sigma\Delta$ modulators. We confirm the predicted gain of 3$dB$/octave over the classical reconstruction, even for modulators of order greater than 2, where the projection on $\Gamma(C_0)$ performed by our algorithm is only approximate. We also show that performing a reconstruction in $V_0 \cap \Gamma(C_0)$ is particularly insensitive to circuit nonidealities (provided they are known): no significant loss in SNR is observed even with nonidealities of 1% on the quantization thresholds and on the feedback D/A outputs of a $\Sigma\Delta$ modulator.

Assumptions and terminology

Since input signals are assumed to be sampled at a frequency higher than the Nyquist rate, most of the time we will directly deal with their discrete time version and consider them as sequences of real numbers. Therefore, whenever we use the term “analog signal”, we will refer to such sequences. If $X$ is an analog signal, $X(k)$ will designate its $k^{th}$ sample. Although bandlimited signals have an infinite support in time, they can only be known and processed from an initial time. Sequences $X$ will be therefore defined for $k \geq 1$ and $k$ will be called the time index. The space of real sequences with positive time indices will be called $\mathcal{V}$. The
canonical norm \( \| \cdot \|_v \) of \( V \) will be the mapping

\[
X \in V \mapsto \| X \|_v = \left( \sum_{k \geq 1} |X(k)|^2 \right)^{1/2}
\]

When considering periodic signals sampled on \( M \) points, it will be understood that \( V \) is the space of \( M \) point sequences and

\[
\| X \|_v = \left( \sum_{k=1}^M |X(k)|^2 \right)^{1/2}
\]

A digital output \( C \) will be considered as a sequence of codewords \( C(k) \). We will often call it code sequence. We will use the term operator to denote a transformation which maps a sequence into another sequence. The image sequence need not belong to the same space as the original sequence. For example, an A/D converter is an operator which maps real sequences into code sequences. When an operator \( H \) maps \( X \) into \( Y \), we will write \( Y = H[X] \). We will reserve the notation ‘DAC’ for the operator of D/A conversion. For a given A/D converter, if a signal \( X_0 \) gives a code sequence \( C_0 \) the signal cell \( \Gamma(C_0) \) will be also denoted by \( \Gamma(X_0) \), depending on whether the object we are referring to is \( C_0 \) or \( X_0 \).
Chapter 2

Analysis of ADC signal partition

In this chapter, we show how the signal partition generated by most A/D converters currently known can be determined and that it is convex. This will enable us to identify the signal cell corresponding to a given code sequence. We saw in chapter 1 that this is an important step for the generation of estimates giving the same code sequence as that of an original analog signal. We will separate A/D converters into two categories: those which include only one quantizer (single-quantizer converters) and those which include more than one quantizer (multi-quantizer converters). The first category includes, for example, simple, dithered and predictive ADC, single-loop or multi-loop ΣΔ modulators and interpolative modulators. The second category includes for example multi-stage modulators.

We start studying signal partitioning for the first category (sections 2.1 and 2.2). We will see that the convexity of the signal partition generated by such converters is a direct consequence of the inherent convexity of a quantizer. In section 2.3, we show how this property can be used for signal reconstruction improvement. In section 2.4, we show that it is in fact possible to improve an estimate by using only a partial knowledge of the code sequence. We finally see in section 2.5 how these properties can be applied to multi-quantizer converters.

2.1 Simple ADC

The key function of any A/D converter is the quantizer. It is the device which, at every sampling instant, receives an analog input sample and outputs a codeword (in practice a binary number). Rigorously speaking, this code identifies that the input sample belongs to a certain interval, called quantization interval, inside a predefined subdivision of the input amplitude range.

According to our terminology, simple ADC is the type of conversion which is reduced to a single quantizer (figure 2.1). Figure 2.2a shows an example of conversion of an M point analog sequence X. The output is an M point sequence of codewords and is determined by the definition of the quantization subdivision. We have chosen on purpose letters for the codewords, so that one is not tempted to interpret C as a quantized signal. The signal cell Γ(C) can be determined as soon as the quantization subdivision is known, or equivalently, the quantization thresholds. For example, any signal giving the code sequence of figure 2.2a is necessarily confined between the arrows of figure 2.2b. This identification of a signal cell in simple ADC is therefore straightforward. This takes circuit imperfections into account if they are included in the definition of the quantization thresholds.
2.2 Single-quantizer conversion

A very large family of ADC falls into what we call “single-quantizer conversion”. It includes every converter which uses only one quantizer. We have found that most of the time, such a converter can be modeled by the structure shown in figure 2.4, where $H$ is an invertible affine operator on analog sequences, and $G$ a given operator with digital input and analog output. By affine, we mean that $H$ can be written $H[X] = L[X] + L_0$ where $L$ is a linear operator and $L_0$ is a constant signal. In practice, $G$ includes a D/A converter (DAC).

Let us go through an exhaustive list of A/D converters and see how they can be reduced to figure 2.4. Simple ADC is the particular case where $H = I$ (identity operator) and $G = 0$ (null operator). Dithered ADC is modeled by considering $H = I$ and $G$ as the generator of the dither signal. In this case, $G$ does not depend on its input. Simple $\Delta$ modulation is the case where $H = I$ and $G$ is simply a DAC. In general, any kind of predictive ADC can be reduced to figure 2.4 by taking $H = I$. The definition $G$ will depend on the considered predictive modulator. An interpolative modulator which is shown in figure 2.5 can be equivalently represented by figure 2.4 by keeping the same operator $H$, and defining $G$ as the composition of a DAC, a delay and the filter $H$. The operator $H$ will be invertible and affine as soon as the initial conditions of filtering are known. As a last example, let us deal with the $n^{th}$ order $\Sigma\Delta$ modulator, shown in figure 2.6: its structure is equivalent to figure 2.4 when taking $H$ and $G$ shown in figures 2.7a-b respectively. As soon as the initial states of the successive accumulators of $H$ are known, $H$ can be expressed $H[X] = L[X] - L_0$ where $L$ is the linear and invertible operator shown in figure 2.7c and $L_0$ is the constant signal obtained from figure 2.7d. The operator $L$ is simply an $n^{th}$ order discrete integrator.

**Remark:** The $n^{th}$ order $\Sigma\Delta$ modulator is often represented as in figure 2.8, but this is equivalent to figure 2.6 provided we define $C'(k) = C(k - 1)$. Our modified structure of figure 2.6 will be convenient for the design of the code projection, and is always possible as the time shift on the code sequence can easily be realized in postprocessing.

Now that we know how to reduce a given A/D converter to figure 2.4 let us use this
Figure 2.2: (a) General principle of quantization. The amplitude axis is divided into predefined quantization intervals. Each of them is labeled by a codeword ('a' to 'e'). The quantizer maps a signal $X$ into a sequence $C$ of codewords. (b) Time representation of the signal cell $\Gamma(C)$. The code sequence $C$ indicates at every time index $k$ in what quantization interval the sample of a signal in $\Gamma(C_0)$ should belong to.
Figure 2.3: Geometric representation of the signal partition for 2 point sequences with the quantizer of figure 2.2a.

equivalent structure to identify the signal cell corresponding to a code sequence $C$. The first step is to see on figure 2.4 that $C$ is the simple A/D conversion of signal $A$. From the previous paragraph, we know how to identify the signal cell of signal $A$ covered by a fixed code sequence $C_0$. A time representation of this cell still has the form of figure 2.2b and depends directly on the quantization subdivision. To deduce the signal cell in the space of $X$, we first point out that, for any signal $A$ giving code sequence $C_0$, the output of $G$ will always be the fixed signal $D_0 = G[C_0]$. Therefore, the signal cell in $X$ is generated by $H^{-1}[A + D_0]$ for every $A$ giving $C_0$ through the quantizer.

Several remarks can be made. If nonlinearities exist in the feedback loop of a converter, they can be taken into account in the definition of $G$ and therefore in the calculation of $D_0 = G[C_0]$. Once $D_0$ is known, we can consider the virtual converter represented in figure 2.9. This encoder is of course different from the original one since it has no feedback but a

Figure 2.4: General structure of a single-quantizer converter: $H$ is an invertible affine operator and $G$ a digital to analog operator.
dither $D_0$ instead. Therefore, it generates a different signal partition. Let us call $\Gamma'(C_0)$ the signal cell of $C$ for the virtual converter. However, since $D_0 = G[C_0]$, it is easy to see that, for the particular case of the code sequence $C_0$, we have $\Gamma(C_0) = \Gamma'(C_0)$.

**Fact 2.1** Consider a single-quantizer converter (figure 2.4), a fixed code sequence $C_0$ and a converter of the type of figure 2.9 with $D_0 = G[C_0]$. The signal cells of $C_0$ generated by the two converters are the same. The second converter will be called the virtual converter associated with the first converter and code sequence $C_0$.

In the following, once a fixed code sequence $C_0$ is considered, it will be very convenient to work with the virtual converter.

### 2.3 Convexity of signal cells and projection

We are going to show that every converter mentioned above have convex signal cells. This is straightforward in the case of simple ADC. We saw in paragraph 2.1 that signal cells are hyper parallelepipeds in space $\mathcal{V}$. Therefore, they are convex. For a single-quantizer converter of general type, we saw that $\Gamma(C_0)$ is generated by signals of the form $H^{-1}[A + D_0]$ where $A$ yields the code sequence $C_0$ through the quantizer and $D_0 = G[C_0]$. Signals $A$ necessarily belong to a convex set (more precisely a parallelepiped), and $\Gamma(C_0)$ is therefore the transform of this convex set through the affine transform $A \mapsto H^{-1}[A + D_0]$. Thus, $\Gamma(C_0)$ is convex.

Another way to prove convexity is to deal with virtual converters. To start with, it is easy to prove directly the following fact:
Figure 2.7: (a-b) Equivalent $H$ and $G$ operators of an $n^{th}$ order $\Sigma\Delta$ modulator. $H$ is obtained by calculating the signal $A$ of figure 2.6 when the feedback DAC output is replaced by zero. $G$ is obtained by calculating the signal $-A$ of figure 2.6 when the input is replaced by zero. (c) Linear part $L$ of $H$, obtained by forcing the initial state of every accumulator to be zero. (d) Constant part $L_0$ of $H$, obtained by taking the output of $H$ when the input is zero.
Figure 2.8: Usual representation of an \( n^{th} \) order \( \Sigma \Delta \) modulator. It is equivalent to figure 2.6 when taking \( C'(k) = C(k - 1) \).

Figure 2.9: Reduced coding structure: \( H \) is an invertible affine operator and \( D_0 \) a known sequence. This leads to the virtual converter of figure 2.4 for \( C_0 \) when \( D_0 = G[C_0] \).

**Fact 2.2** Signal cells generated by converters of the type of figure 2.9 are convex.

Proof: We have to show that if \( X_0, X_1 \) are two signals giving the same code sequence \( C \), then any signal of the form \( X_\theta = (1 - \theta)X_0 + \theta X_1 \) where \( \theta \in [0, 1] \), should also give \( C \). Since \( H \) is affine, it can be written \( H[X] = L[X] + L_0 \), where \( L \) is a linear operator and \( L_0 \) is a fixed sequence. Let \( A_\theta, A_0 \) and \( A_1 \) be the signals seen by the quantizer when \( X, X_0, X_1 \) are input respectively. We have

\[
A_0 = L[X_0] + (L_0 - D_0) \quad A_1 = L[X_1] + (L_0 - D_0) \quad A_\theta = L[X_\theta] + (L_0 - D_0)
\]

Using the linearity of \( L \) and the fact that \( 1 = (1 - \theta) + \theta \), we have:

\[
A_\theta = (1 - \theta)(L[X_0] + (L_0 - D_0)) + \theta L[X_1] + (1 - \theta + \theta)(L_0 - D_0)
\]

\[
= (1 - \theta)(L[X_0] + (L_0 - D_0)) + \theta(L[X_1] + (L_0 - D_0)) = (1 - \theta)A_0 + \theta A_1
\]

By assumption, at every time index \( k \), \( A_0(k) \) and \( A_1(k) \) belong to the same quantization interval: the one identified by \( C(k) \). This is also true for \( A_\theta(k) = (1 - \theta)A_0(k) + \theta A_1(k) \) which belongs to the interval \([A_0(k), A_1(k)]\), and this for every time index \( k \). Therefore \( X_\theta \) gives the code sequence \( C \) \( \square \)

The convexity of signal cells of any single-quantizer converter follows using their equivalence to figure 2.9 and this fact.

As already mentioned in chapter 1, convexity provides us with a fundamental tool for signal reconstruction. Without convexity, it did not seem too obvious why it is interesting to
look for an estimate in $\Gamma(C_0)$. For example, this approach would seem inadequate if $\Gamma(C_0)$ was not connected (in one piece, roughly speaking). With convexity, we not only have the guarantee that $\Gamma(C_0)$ is connected, but we have a tool to (at least theoretically) improve any proposed estimate $X$ which does not belong to the signal cell containing $X_0$ namely the projection on $\Gamma(C_0)$. By definition, the projection of $X$ on $\Gamma(C_0)$ is the signal $X' \in \Gamma(C_0)$ which minimizes the distance with $X$. Because $\Gamma(C_0)$ is convex, it is a consequence of Hilbert space properties that the distance between $X'$ and $X_0$ is necessarily inferior to that between $X$ and $X_0$.

In chapter 4 we will actually propose algorithms to compute the projection of $X$ on $\Gamma(C_0)$ in the case of simple, dithered, predictive ADC and the 1st order $\Sigma\Delta$ modulation.

2.4 Signal cell of a code subsequence, convexity and projection

What really served us in the improvement of an estimate is the fact that $\Gamma(C_0)$ is a convex set that we can identify (because we know $C_0$) and which contains $X_0$ for sure. Actually, this principle of estimate improvement will work as soon as we can identify a convex set which contains $X_0$, whatever the method is, provided it is based on the available information. In this section, we prove that a partial knowledge of $C_0$, more precisely, the knowledge of $C_0$ on a subset $I$ of time indices, is enough to determine a convex set which necessarily contains $X_0$. The search for such convex set will be very helpful in the future, especially when trying to perform convex projections. Indeed, when the order of a converter becomes larger than 2, it will become unrealistic to compute the projection of an estimate, taking into account the sequence $C_0$ on its full time range (chapter 4).

The first attempt is to define $\Gamma_I(C_0)$ as the set of all signals giving code sequences which coincide with $C_0$ only on the time subrange $I$:

$$\Gamma_I(C_0) = \{ X \in \mathcal{V} \mid \text{if } X \text{ gives code } C, \forall k \in I, C(k) = C_0(k) \}$$

It is obvious that $\Gamma_I(C_0)$ contains $\Gamma(C_0)$ and therefore includes $X_0$. More precisely, we have the logic relation:

if $I_1 \subset I_2$ then $\Gamma_{I_1}(C_0) \supset \Gamma_{I_2}(C_0) \supset \Gamma(C_0) \ni X_0$

Qualitatively speaking, the closer $I$ will be to the full time range, the closer $\Gamma_I(C_0)$ will be to $\Gamma(C_0)$. Unfortunately, as suggested by our expression “first attempt”, if $I$ does not cover the full time range, there is no reason for $\Gamma_I(C_0)$ to be convex in general. The situation is not hopeless.

Once the code sequence $C_0$ is fixed, we can always consider the associated virtual converter of figure 2.9, which has the same signal cell $\Gamma(C_0)$. Let us call $\Gamma'_I(C_0)$ the set of signals giving code sequences which coincide with $C_0$ on $I$ for the virtual converter. There is no reason to have $\Gamma_I(C_0) = \Gamma'_I(C_0)$, but we will still have:

if $I_1 \subset I_2$ then $\Gamma'_{I_1}(C_0) \supset \Gamma'_{I_2}(C_0) \supset \Gamma'(C_0) = \Gamma(C_0) \ni X_0$

Then we have the following fact:

To be rigorous, the projection $X'$ exists (and is unique) only if $\Gamma(C_0)$ is also topologically closed. Actually, we will always consider the projection $X'$ on $\Gamma(C_0)$, the closure of $\Gamma(C_0)$. Such a signal $X'$ will not necessarily belong to $\Gamma(C_0)$, but qualitatively speaking, will be at the limit to belong to it. Nevertheless, the property we effectively need is that $X'$ is a better estimate of $X_0$ than $X$. 

17
signal cells for original converter

Figure 2.10: Geometric representation of the relative positions between signal cells of original and virtual converters. \( \Gamma'_1(C_0) \) and \( \Gamma'_2(C_0) \) are necessarily convex. The larger \( I \) is, the closer \( \Gamma'_1(C_0) \) will be to \( \Gamma(C_0) \) and the closer the projection of \( X \) on \( \Gamma'_1(C_0) \) will be to \( X_0 \).

**Fact 2.3** Let us consider a converter of the type of figure 2.9, a fixed code sequence \( C \) and a fixed time index subset \( I \). Then, the set of signals whose digital outputs coincide with \( C \) on \( I \), is convex.

Proof: It is similar to that of fact 2.2. The difference is that we start with \( X_0, X_1 \) giving code sequences \( C_0, C_1 \) such that \( \forall k \in I, C_0(k) = C_1(k) = C(k) \). Using the same notation as in fact 2.2, the equality \( A_\theta = (1-\theta)A_0 + \theta A_1 \) is still true since this is a direct consequence of the structure of the converter. This time, if \( k \not\in I \), we cannot draw any conclusion about the position of \( A(k) \). On the other hand, if \( k \in I, A_0(k) \) and \( A_1(k) \) belong to the same quantization interval, namely the interval labeled \( C(k) \). This is necessarily the case for \( A_\theta(k) = (1-\theta)A_0(k) + \theta A_1(k) \). Therefore, the code sequence produced by \( X \) coincides with \( C \) on \( I \) \( \square \)

The set \( \Gamma'_1(C_0) \) has no longer any intuitive interpretation but it has the following properties:
- \( \Gamma'_1(C_0) \) includes \( \Gamma(C_0) \) which itself contains \( X_0 \)
- \( \Gamma'_1(C_0) \) is convex

Therefore, projecting an estimate \( X \) of \( X_0 \) on \( \Gamma'_1(C_0) \) will necessarily improve it. The closer \( I \) will be to the full time range, the closer the projection on \( \Gamma'_1(C_0) \) will be to the projection on \( \Gamma(C_0) \), but also to \( X_0 \) itself. Figure 2.10 recapitulates the situation between the different signal cells and projections. Note that, it is not quite correct to say that \( \Gamma'_1(C_0) \) takes into account only a partial knowledge of \( C_0 \); the dither sequence \( D_0 \) computed for the virtual converter does depend on the full code sequence \( C_0 \).

From now on, as soon as we consider signal cells covered by a code subsequence, we will automatically refer to the virtual converter and use the notation \( \Gamma(C_0) \) without 's. In fact, we will consider the case \( I \) equal to the full time range as a particular case of convex set
Figure 2.11: (a) General structure of a multi-stage converter; (b) Structure considered in this paper. The arithmetic stage included in the structure of (a) is a first step to signal reconstruction: we do not consider it as a part of the A/D converter.

\( \Gamma_I(C_0) \): this is the smallest convex set derivable from \( C_0 \) containing \( X_0 \). But, more important, it coincides with \( \Gamma(C_0) \), the set of all signals giving \( C_0 \) through the original converter. Therefore, when we deal with code projection in chapter 4, we will automatically work with the transformed virtual converter. This will enable us to design projection algorithms for converters of order higher than 2 (\( n^{th} \) order multi-loop, multi-stage \( \Sigma \Delta \) modulators)

### 2.5 Multi-quantizer conversion

We call “multi-quantizer converter” any A/D converter which includes more than one quantizer. We only consider the important example of multi-stage modulators, shown in figure 2.11a.

The quantization error is output from every single-quantizer converter in the way shown in figure 2.12, where the DAC operation has been extracted from the operator \( G \). Multi-stage \( \Sigma \Delta \) modulation is the particular case where every subconverter is a \( \Sigma \Delta \) modulator of the type of figure 2.6. Note that we consider the arithmetic stage to be a part of the reconstruction process, and therefore, for our reconstruction purposes, we want to consider directly the digital sequences entering the arithmetic stage. This was also done by Hein and Zakhor [12]. The arithmetic stage is designed under the assumption of white quantization noise. Here, we take the “real” digital sequence of the multi-stage modulator, which is the \( p \)-tuple of code sequences \( (C_1, \ldots, C_p) \) (figure 2.11b). Note that, for the very common case of two-stage single-bit modulation, the digital output is the same (2bits/sample) whether an output arithmetic operation is included in the converter or not.

Unlike in the previous paragraph, we will not try to find a general structure for multi-
quantizer converters. For the analysis of signal cells, we are going to show directly that, for a given code sequence \( C_0 = (C_{01}, C_{02}, ..., C_{0p}) \), it is possible to build a virtual converter with the structure of figure 2.13 which has exactly the same signal cell \( \Gamma(C_0) \). Let us show this on the simplest case of a two-stage modulator. The detailed structure is shown in figure 2.14a. Let us consider a fixed output code sequence \( C_0 = (C_{01}, C_{02}) \). From figure 2.14a, we derive a virtual converter by forcing the input of the \( i^{th} \) DAC to be equal to \( C_{0i} \) (figure 2.14b). Similarly to fact 2.1, one has to be convinced that the two converters yield exactly the same signal cell \( \Gamma(C_0) \), even if in general they do not define the same partition in \( \mathcal{V} \). The second step is to see that the structure of figure 2.14b is equivalent to figure 2.14c when taking \( H_i' = H_1, \ H_i'' = -H_2 \circ H_1 \) and signals \( D_{0i1}, D_{0i2} \) computed from \( C_{0i1}, C_{0i2} \) as shown in figures 2.14d and 2.14e respectively. This principle of transformation can be applied to the case of \( p \) quantizer modulators. The \( i^{th} \) operator \( H_i' \) of the virtual converter (figure 2.13) will be equal to

\[
H_i' = (-1)^{i-1} H_i \circ H_{i-1} \circ \cdots \circ H_1
\]

if \( H_i \) is the \( i^{th} \) operator of the \( i^{th} \) single-path modulator in the original converter (figure 2.11).

Now, using the structure of figure 2.13, we say that \( \Gamma(C_0) = \Gamma_1(C_{01}) \cap ... \cap \Gamma_p(C_{0p}) \) where \( \Gamma_i(C_{0i}) \) is the signal cell covered by \( C_{0i} \) for the \( i^{th} \) subconverter. From paragraph 2.2, we know how to generate every single \( \Gamma_i(C_{0i}) \). Unlike the case of single-quantizer conversion in section 2.2, we will not be able to propose an analytical generation of signals belonging to \( \Gamma(C_0) \). However, we know how to identify from the virtual structure of figure 2.13 the signal cell \( \Gamma_i(C_{0i}) \) for the \( i^{th} \) converter. A signal will belong to \( \Gamma(C_0) \) if and only if it belongs to every \( \Gamma_i(C_{0i}) \) for \( i = 1, ..., p \). Therefore

\[
\Gamma(C_0) = \Gamma_1(C_{01}) \cap \Gamma_2(C_{02}) \cap \cdots \cap \Gamma_p(C_{0p})
\]

Since \( \Gamma_i(C_{0i}) \) is convex, \( \Gamma(C_0) \) is necessarily convex.

If \( X_0 \) is an analog signal giving the code sequence \( C_0 \), the theoretical idea of projecting an estimate \( \hat{X} \) of \( X_0 \) on \( \Gamma(C_0) \) to improve it, is still valid. In practice, since we have no analytical description of \( \Gamma(C_0) \), we will not propose an algorithm for the direct projection on \( \Gamma(C_0) \). However, \( X \) can be improved by successive and periodic projections on \( \Gamma_1(C_{01}), \Gamma_2(C_{02}), ..., \Gamma_p(C_{0p}) \). Youla [33] has actually shown that such iteration of projections converges and leads to an element of the intersection \( \Gamma(C_0) \).

Coming back to the case \( p = 2 \), it can be seen from figures 2.14d and 2.14e that nonlinearities in the feedback DAC outputs if they are measurable are automatically included in
the definition of $D_{01}$ and $D_{02}$. In general, internal feedback imperfections will be included in the dither sequences $D_{01},\ldots, D_{0p}$ of the virtual structure (figure 2.13).

In conclusion, we have shown that, by structure transformation, the analysis of a multi-quantizer converter is reduced to the study of single-quantizer converters.

### 2.6 Conclusion

We showed how the signal partition generated by most A/D converters currently known can be derived. This partitioning analysis consists in identifying the set $\Gamma(C)$ of all possible signals producing a given digital sequence $C$, for a given A/D converter. The definition of the partition only depends on the direct description of the converter’s structure, and includes circuit imperfections if they can be measured. The convexity of such partition is a direct consequence of the inherent convexity of a quantizer. This analysis provides us with theoretical means to improve any estimate which does not reproduce the same digital sequence as that of a given original analog signal, using orthogonal projections. Practical algorithms which perform such projections will be actually derived in chapter 4.

Basically, the signal partitioning effect of an A/D converter is a consequence of quantization. The convexity property of a quantizer simply results from the fact that quantization intervals are naturally convex themselves. This approach of convex coding analysis can be generalized to vector quantization.
Figure 2.14: (a) General structure of a two-quantizer converter. (b) Virtual version corresponding to code sequence $C_0 = (C_{01}, C_{02})$. The signal cell $C_0$ is the same as that of the original converter (a). (c) Virtual converter equivalent to (b). (d) Computation of the dither sequence $D_{01}$ of (c). It is the value of the signal $-A_1$ of (b) when $X$ is replaced by zero. (e) Computation of the dither sequence $D_{02}$ of (c). It is the value of the signal $-A_2$ of (b) when $X$ is replaced by zero.
Chapter 3

MSE analysis of optimal signal reconstruction in oversampled ADC

In this chapter, we suppose that input signals belong to the subspace $V_0 \subset \mathcal{V}$ of sequences bandlimited to a known maximum frequency $f_m$: this is the situation of oversampled ADC. Once a signal $X_0 \in V_0$ is given by a code sequence $C_0$ output by a given A/D converter, we propose to study the estimation of $X_0$ which consists in “picking up” a signal inside $V_0 \cap \Gamma(C_0)$. Theoretically, it is possible to reach such a signal by alternating projections of a first estimate on $V_0$ and $\Gamma(C_0)$ (Youla’s theorem [3]). As said in chapter 1, we will propose algorithms which actually perform these projections, in chapter 4.

We study MSE bounds of such estimates, first for the simple ADC and then for any single-quantizer converter where the operator $H$ is a cascade of integrators (figure 2.4). This includes, for example, the $n^{th}$ order multi-loop $\Sigma\Delta$ modulator. As a consequence, we will see that MSE bounds can be deduced for multi-stage, multi-loop $\Sigma\Delta$ modulators as well. Hein and Zakhov measured MSE upper bounds for the $1^{st}$ and $2^{nd}$ order $\Sigma\Delta$ modulators in the particular case of dc input (one dimensional $V_0$) [9, 11].

In our MSE bound evaluation, we will not make any assumption about how the estimate $X$ of $X_0$ is chosen inside $V_0 \cap \Gamma(C_0)$. Thus our analysis will be applicable to any reconstruction method leading to a reconstructed signal inside $V_0 \cap \Gamma(C_0)$.

3.1 Modelization of bandlimited and periodic signals ($V_0$)

Continuous time bandlimited periodic signals of period $T_0$ can be expressed as follows:

$$X(t) = A + \sum_{j=1}^{N} B_j \sqrt{2} \cos(2\pi j \frac{t}{T_0}) + \sum_{j=1}^{N} C_j \sqrt{2} \sin(2\pi j \frac{t}{T_0})$$

(3.1)

Such signals have $2N+1$ discrete low frequency components, between $-\frac{N}{T_0}$ and $\frac{N}{T_0}$ in the continuous Fourier domain. They can be equivalently described by their sampled version:

$$X(k) = X \left[ \frac{k}{M} T_0 \right] = A + \sum_{j=1}^{N} B_j \sqrt{2} \cos(2\pi j \frac{k}{M}) + \sum_{j=1}^{N} C_j \sqrt{2} \sin(2\pi j \frac{k}{M})$$

(3.2)

23
when choosing the sampling frequency of the form $\frac{M}{2^M}$ such that $M > 2N + 1$ (oversampling situation). In this section, we will consider that $V_0$ is the space of sequences of the form (3.2) for a fixed $N$, with the assumption that $M > 2N + 1$. Note that the oversampling rate $R$ is proportional to $M$. An element of $V_0$ given by (3.2) (or (3.1)) is uniquely defined by the vector $\tilde{X} \in \mathbb{R}^{2N+1}$ whose components are $(A, B_1, \ldots, B_N, C_1, \ldots, C_N)$. In other words, we have an isomorphism between $V_0$ and $\mathbb{R}^{2N+1}$. The dimension of $V_0$ is therefore $2N + 1$. Using Parseval’s equality, it can be derived that

$$\frac{1}{M} \sum_{k=1}^{M} |X(k)|^2 = A^2 + \sum_{j=1}^{N} B_j^2 + \sum_{j=1}^{N} C_j^2 = \|\tilde{X}\|^2$$

where $\|\tilde{X}\|$ is the canonical norm in $\mathbb{R}^{2N+1}$. Therefore, if $X \in V_0 \cap \Gamma(X_0)$ is an estimate of $X_0 \in V_0$, the mean squared error $MSE(X_0, X)$ between $X_0$ and $X$ can be measured by the distance in $\mathbb{R}^{2N+1}$ between their associated vectors $\tilde{X}_0$ and $\tilde{X}$:

$$MSE(X_0, X) = \frac{1}{M} \sum_{k=1}^{M} |X(k) - X_0(k)|^2 = \|\tilde{X} - \tilde{X}_0\|^2$$  \hspace{1cm} (3.3)

We will use this isometry between $V_0$ and $\mathbb{R}^{2N+1}$ throughout this section. We will in particular try to measure the asymptotic evolution of this MSE bound with increasing oversampling rate. Since $M$ is proportional to $R$, we will compare the MSE with powers of $M$.

### 3.2 Simple ADC

We recall that for simple ADC, the classical signal reconstruction yields a gain in SNR of $3dB$ per octave of oversampling. Increasing $R$ by an octave means doubling the resolution in time of the signal discretization. It is interesting to see that when the resolution in amplitude is doubled, the gain is not $3dB$ but $6dB$ (in short, the gain is $6dB$ per bit). In theory, one would like to think ADC as a homogeneous discretization of a two dimensional graph, where the dimensions are amplitude and time. One could argue that the origin of this difference is the non-homogeneity of the error measurement: the MSE is a measure of errors only along the amplitude dimension. However, a bandlimited signal has a bounded slope. Therefore, qualitatively speaking, variations in the time dimension should be observable in the amplitude dimension with a bounded multiplicative coefficient which should not affect the logarithmic dependence of the MSE. These considerations give another hint that the classical signal reconstruction in simple ADC is not optimal.

In the theorem we are about to present, we show that, in the context of periodic signals, an estimate $X$ of $X_0$ chosen in $V_0 \cap \Gamma(X_0)$ will yield an MSE which goes to zero at least as fast as $\frac{2^\alpha}{M^2}$ when $M$ goes to $+\infty$, where $\alpha$ is a constant independent of $M$: this actually represents a gain of $6dB$ per octave of oversampling. This property is however verified under a certain condition about the source signal $X_0$: it must cross the quantization thresholds at least $2N + 1$ times. This condition relative to the amplitude dimension is in fact the dual condition to the restriction in the time dimension that $M$ should be larger than $2N + 1$. If the last condition fails, aliasing will result from sampling and introduce an irreversible error even before the signal is quantized in amplitude. Obviously, the $6dB$/bit gain should not be expected in this case. Similarly, if the number of threshold crossings is less than $2N + 1$, the gain of $6dB$ per octave of oversampling should not be expected either.

24
Theorem 3.1 If the continuous time version of a signal $X_0 \in V_0$ has more than $2N + 1$ quantization threshold crossings\(^1\), there exists a constant $c_0$ which only depends on $X_0$ such that, for $M$ large enough and any $X$ in $V_0 \cap \Gamma(X_0)$, $MSE(X_0, X) \leq \frac{c_0}{M^2}$.

Proof: The set of instants where $X_0[t_i]$ is equal to one of the threshold levels is discrete and finite (since $X_0$ cannot be constant). Let $\Delta t^0 > 0$ be the minimum distance between these instants. We consider $M$ large enough (or $T_s = \frac{T_0}{2}$ small enough) so that $T_s < \frac{\Delta t^0}{3}$. Let us choose $2N + 1$ distinct threshold crossings of $X_0[t_i]$, call $(t_0^0, ..., t_{2N}^0)$ their instants and $(l_1, ..., l_{2N})$ their levels.

We are first going to show that any signal $X \in V_0 \cap \Gamma(X_0)$ necessarily crosses level $l_i$ at an instant $t_i$ in the same sampling interval as that of $t_i^0$, for every $i = 0, ..., 2N$. Let $(k_0, ..., k_{2N})$ be the time indices such that the sampling interval $I_i = [(k_i - 1)T_s, k_i T_s]$ contains $t_i^0$ for every $i = 0, ..., 2N$. For a fixed $i$, $X_0[t_i]$ crosses level $l_i$ which is the common boundary of two quantization intervals. Because $T_s < \frac{\Delta t^0}{3}$, $X_0(k_i - 1)$ and $X_0(k_i)$ necessarily belong respectively to the interior of these two intervals. If $X \in V_0 \cap \Gamma(X_0)$, $X(k_i - 1)$ and $X(k_i)$ should respectively belong to the quantization intervals which contain $X_0(k_i - 1)$ and $X_0(k_i)$. Therefore, since it is continuous, $X[t]$ should cross level $l_i$ at a time $t_i \in I_i$. As a consequence, we have

$$|t_i - t_i^0| \leq T_s = \frac{T_0}{M} \quad (3.4)$$

Since $T_s < \frac{\Delta t^0}{3}$, we also have

$$\forall i_1, i_2 \in \{0, ..., 2N\} \text{ such that } i_1 \neq i_2, \quad |t_{i_1} - t_{i_2}| > \frac{\Delta t^0}{3} \quad (3.5)$$

Using (3.4), we are going to show that the norm of $\vec{X} - \vec{X}_0$ can be linearly bounded with $\frac{1}{M}$ when $M$ goes to $+\infty$. This will be based on a linearization of variations in time and an algebraic manipulation.

Using (3.1), the fact that $X[t_i] = X_0[t_i^0] = l_i$ for every $i = 0, ..., 2N$ can be written with a vector notation as

$$\mathcal{M}(t_0^0, ..., t_{2N}^0) \vec{X} = \mathcal{M}(t_0^0, ..., t_{2N}^0) \vec{X}_0 = \vec{L} \quad (3.6)$$

where $\mathcal{M}(s_0, ..., s_{2N})$ is the transpose of the square matrix

$$
\begin{pmatrix}
1 & \cdots & 1 \\
\sqrt{2} \cos(2\pi s_0 / T_0) & \cdots & \sqrt{2} \cos(2\pi s_{2N} / T_0) \\
\vdots & \cdots & \vdots \\
\sqrt{2} \cos(2\pi N s_0 / T_0) & \cdots & \sqrt{2} \cos(2\pi N s_{2N} / T_0) \\
\sqrt{2} \sin(2\pi s_0 / T_0) & \cdots & \sqrt{2} \sin(2\pi s_{2N} / T_0) \\
\vdots & \cdots & \vdots \\
\sqrt{2} \sin(2\pi N s_0 / T_0) & \cdots & \sqrt{2} \sin(2\pi N s_{2N} / T_0) \\
\end{pmatrix}
$$

\(^1\)We will not count as threshold crossings, points where $X_0[t]$ reaches a threshold without crossing it. As a consequence, $X_0$ cannot be a constant.
and $\bar{L}$ the column vector whose components are $(l_0, ..., l_{2N})$. Replacing the cos and sin functions in $\mathcal{M}$ by their complex exponential expressions, it is possible to see that $\mathcal{M}(s_0, ..., s_{2N})$ is similar to the Vandermonde matrix

$$[\exp(j2\pi ks_i/T_0)]_{0 \leq i \leq 2N, -N \leq k \leq N}$$

which is invertible as soon as $(s_0, ..., s_{2N})$ are distinct. Therefore, $\mathcal{M}(t_0, ..., t_{2N})$ and $\mathcal{M}(t_0^0, ..., t_{2N}^0)$ are invertible. It can also be shown that, because of (3.5), $[\mathcal{M}(t_0^0, ..., t_{2N}^0)]^{-1}$ is bounded. Therefore, $\bar{X} = [\mathcal{M}(t_0^0, ..., t_{2N}^0)]^{-1} \bar{L}$ is bounded. Subtracting $\mathcal{M}(t_0^0, ..., t_{2N}^0) \bar{X}$ in the two first members of equation (3.6), we find

$$\mathcal{M}(t_0^0, ..., t_{2N}^0)(\bar{X} - \bar{X}_0) = - \left( \mathcal{M}(t_0, ..., t_{2N}) - \mathcal{M}(t_0^0, ..., t_{2N}^0) \right) \bar{X}$$

(3.7)

At the limit of $M$ going to $+\infty$, $t_i - t_i^0$ goes to zero and we have

$$\mathcal{M}(t_0, ..., t_{2N}) - \mathcal{M}(t_0^0, ..., t_{2N}^0) \simeq \sum_{i=0}^{2N} (t_i - t_i^0) \frac{\partial \mathcal{M}}{\partial \bar{t}_i}(t_0^0, ..., t_{2N}^0)$$

(3.8)

Because $\bar{X}$ is bounded and $\mathcal{M}(t_0^0, ..., t_{2N}^0)$ is invertible, (3.7) and (3.8) imply that

$$\bar{X} - \bar{X}_0 \simeq - [\mathcal{M}(t_0^0, ..., t_{2N}^0)]^{-1} \sum_{i=0}^{2N} (t_i - t_i^0) \frac{\partial \mathcal{M}}{\partial \bar{t}_i}(t_0^0, ..., t_{2N}^0) \bar{X}$$

(3.9)

Since $\bar{X}$ is bounded, the right hand side goes to zero when $M$ goes to $+\infty$. Therefore, $\bar{X}$ tends to $\bar{X}_0$, and (3.9) is still true when replacing $\bar{X}$ by $\bar{X}_0$ in the right hand side (since $\bar{X}_0 \neq \bar{0}$). We obtain

$$\bar{X} - \bar{X}_0 \simeq \sum_{i=0}^{2N} (t_i - t_i^0) \bar{F}_i$$

where

$$\bar{F}_i = - [\mathcal{M}(t_0^0, ..., t_{2N}^0)]^{-1} \frac{\partial \mathcal{M}}{\partial \bar{t}_i}(t_0^0, ..., t_{2N}^0) \bar{X}_0$$

is a vector which depends only on the signal $X_0$. Using (3.4), we find $||\bar{X} - \bar{X}_0|| \leq \bar{c}$ where $\bar{c} = T_0 \sum_{i=0}^{2N} ||\bar{F}_i||$. The proof is completed using (3.3) and taking $c_0 = \bar{c}^2$.

Remark : The only assumption used in this theorem was the number of threshold crossings of the input signal. For example, this does not require quantization to be uniform. However, the constant $c_0$ obtained in the upper bound may depend on how the input signal crosses the thresholds. In particular, $c_0$ depends on $F_i$ which contains the term $\frac{\partial \mathcal{M}}{\partial \bar{t}_i}(t_0^0, ..., t_{2N}^0) \bar{X}_0$. One can verify that this expression contains the slope of $X_0$ at the $i^{th}$ threshold crossing.

### 3.3 $n^{th}$ order single-quantizer converter

In theorem 3.1, we needed the condition that the source signal has at least $2N + 1$ threshold crossings within its period. We saw that this is the amplitude dimension dual condition to
the time dimension condition which requires that $M > 2N + 1$. Indeed, $M$ represents the number of “sampling time crossings” of the signal. Since signals are function of time, there is a trivial equality between the number of sampling instants and the number of sampling time crossings. Unfortunately, this is not the case in the amplitude dimension. In general, with rigid quantization thresholds, there is no way to control the number of threshold crossings of a signal, unless particular characteristics are known from it. Then, it becomes natural to think of time varying quantization thresholds. The idea is that, in simple ADC, the efficient information is located around the threshold crossings of the input signal, whereas the rest of the time is just “waiting”. Making the quantization thresholds vary in time is increasing the chances to “intercept” the signal. This is what happens in single-quantizer converters of figure 2.4 when the output of $G$ is not a constant signal. Indeed, if we forget the presence of the operator $H$ for a moment, the code sequence gives the comparison of signal $Y$ with the quantization thresholds shifted by the time varying output of $G$. This last signal can be predefined (dithered ADC) or calculated from the digital output (general case of a single-quantizer converter). In order to evaluate the MSE between $X_0$ and an estimate $X \in V_0 \cap \Gamma(C_0)$, we cannot think in terms of threshold crossings any more: since the variations of the $G$ output can be abrupt, we have lost the notion of bounded slope and cannot use differentiation tools. We have to study how the $G$ output directly affects the definition of $V_0 \cap \Gamma(C_0)$.

To reduce the difficulty of the problem, we are going to confine ourselves to only consider converters where $H$ is an $n^{th}$ order integrator, and the quantizer is uniform with a step size equal to 1 and a finite number of intervals. We naturally include the single-bit quantizer. Moreover, we suppose that the analog signals $X$ to be coded verify a no-overload stability condition that we define as follows: the signal $A$ seen by the quantizer when $X$ is input, never goes farther than a distance equal to 1 from the extreme quantization thresholds. For example, in the case of first order $\Sigma\Delta$ modulation, any signal $X$ where every sample belongs to $[-\frac{1}{2}, \frac{1}{2}]$, will automatically verify the stability condition when the output values of the feedback DAC are $\pm \frac{1}{2}$. In this context, we will assume that an estimate $X_0$ of $X$ in $V_0 \cap \Gamma(X_0)$ should also verify this stability condition. We can include this constraint in the definition of $\Gamma(X_0)$ by considering that the two extreme quantization intervals, which are normally of infinite length, have a size equal to 1 (see figure 3.1). We will then automatically reject as estimate of $X_0$ any signal which gives the same code as $X_0$ but does not verify the stability condition. Finally, in our characterization of $V_0 \cap \Gamma(X_0)$ for a given $X_0$, it will be convenient to consider the virtual converter. This is always possible, since the set of signals giving the same code as $X_0$ is the same whether we look at the original or the virtual converter (fact 2.1). The interesting feature about the virtual converter is the simple characterization of the signal seen by the quantizer. Indeed, when $X$ is input, the signal $A$ appearing in front of the quantizer is $A = H[X] - G[C_0]$ whether $X$ belongs to $\Gamma(X_0)$ or not. Since we want to study the distance or the difference between $X$ and $X_0$, it will be convenient to rewrite this last relation as $A - A_0 = L[X - X_0]$, where $L$ is the linear part of $H$. If we express $X = X_0 + x$, then $A = A_0 + a$ where $a = L[x]$.

Let us first try to characterize the elements of $\Gamma(X_0)$. The action of the $G$ feedback is reflected in the behavior of signal $A_0 = H[X_0] - G[C_0]$. But, to recognize that a signal $X$ belongs to $\Gamma(X_0)$, what really matters is only the relative position of $A_0(k)$ within the quantization interval $Q_0(k)$ it belongs to. Indeed, $X = X_0 + x$ will belong to $\Gamma(X_0)$ if and only if, at every instant $k$, $A_0(k) + a(k)$ and $A_0(k)$ belong to the same quantization interval. If we call $d_+(k)$ and $d_-(k)$ the distance of $A_0(k)$ to the upper and lower bounds of $Q_0(k)$
then,

\[ X_0 + x \in \Gamma(X_0) \iff \forall k = 1, \ldots, M, \quad -d_-(k) \leq a(k) \leq d_+(k), \quad \text{where} \quad a = L[x] \quad (3.10) \]

The two distances \( d_+(k) \) and \( d_-(k) \) are represented in figure 3.1. We always have

\[ d_+(k), d_-(k) \in [0, 1] \quad \text{and} \quad d_+(k) + d_-(k) = 1 \]

Actually, those two numbers are directly related to the quantization error \( e_0(k) \) which is the difference between the center of the interval \( Q_0(k) \) and \( A_0(k) \). From figure 3.1, it is easy to see that:

\[ e_0(k) = -\frac{1}{2} + d_+(k) = \frac{1}{2} - d_-(k) \quad (3.11) \]

Let us now look at what we consider to be the efficient estimates of \( X_0 \): the signals of \( \Gamma(X_0) \) which belong to \( V_0 \). Figure 3.2 shows a 3 dimensional representation of \( \Gamma(X_0) \) and its section with \( V_0 \), symbolized by a 2 dimensional space (since \( V_0 \) has to be a subspace of \( V \)). Any element of \( V_0 \) can be written \( X = X_0 + \lambda U \) where \( \lambda \geq 0 \) and \( U \) belongs to the unit
sphere $S$ of $V_0$: 

$$S = \{ U \in V_0 / \|\tilde{U}\| = 1 \}$$

For an element $U$ of $S$, let us call $V = L[U]$. Then, the equivalence (3.10) applied to elements of $V_0$ becomes

$$X_0 + \lambda U \in \Gamma(X_0) \iff \forall k = 1, \ldots, M, \quad -d_-^0(k) \leq \lambda V(k) \leq d_+^0(k) \quad (3.12)$$

Suppose we fix $U$ and $V = L[U]$, and we only want to look at estimates $X$ of $X_0$ which are located in the direction of $\tilde{U}$ with respect to $X_0$ (see figure 3.2), that is to say, $X$ has the form $X_0 + \lambda U$ where $\lambda \geq 0$. Let us call $\epsilon_k$ the sign of $V(k)$. We can write (3.12) as

$$X_0 + \lambda U \in \Gamma(X_0) \iff \forall k = 1, \ldots, M, \quad \lambda |V(k)| \leq d_{\epsilon_k}^0(k) \quad (3.13)$$

Once we are able to recognize that $X = X_0 + \lambda U \in \Gamma(X_0)$, $MSE(X_0, X)$ will be directly given by $\lambda^2$.

Basically, we have just reduced the characterization of $V_0 \cap \Gamma(X_0)$ to the knowledge of $d_+^0$ or $d_-^0$. Unfortunately, there is no simple expression of $d_+^0$, $d_-^0$ in terms of $X_0$. Even with a statistical approach, Gray [7, 10] showed that, for 1st order $\Sigma\Delta$ modulation with constant inputs, the quantization error signal is not theoretically white. In other words, $d_+^0(k)$ or $d_-^0(k)$ are generally correlated in time.
However, without the knowledge of $d_{+}^0$ or $d_{-}^0$, the mere fact that $A_0(k)$ is confined in an interval of size 1 already gives a first upper bound to $MSE(X_0, X)$ where $X \in V_0 \cap \Gamma(X_0)$. From (3.13), an estimate $X_0 + \lambda U$ in $\Gamma(X_0)$ at least verifies

$$\lambda |V(k)| \leq 1, \forall k = 1, ..., M$$

(3.14)

We recall that $V$ is the result of the $n^{th}$ order integration of $U$. To give an intuition of the upper bound, let us look at the particular direction $\vec{U}$ where $U$ is the unit constant signal: $\forall k = 1, ..., M$, $U(k) = 1$. In this case, without calculation, we will expect $V(k)$ to grow with $k$ in a fashion comparable to $\frac{kn}{M}$. Therefore, $V(k)$ will reach its maximum at $k = M$. From (3.14), we necessarily have $\lambda \leq \frac{1}{\max_{1 \leq k \leq M} |V(k)|} \approx \frac{m}{M}$. We conclude that, if $X \in V_0 \cap \Gamma(X_0)$ differs from $X_0$ by a constant signal, we necessarily have $MSE(X_0, X) \leq \frac{c}{M^{1/2}}$ where $c > 0$ is a constant independent of $M$.

We are going to show that this dependence of the $MSE$ with respect to $M$ is the same with every other direction $U$ of $S$. This time the maximum of $|V(k)|$ is not necessarily achieved at $k = M$, but we still have from (3.14) the necessary condition

$$\lambda \leq \frac{1}{\max_{1 \leq k \leq M} |V(k)|}$$

(3.15)

The following lemma shows that, even if $U$ is not the constant signal, but is in general a periodic and bandlimited signal of energy 1, then $\max_{1 \leq k \leq M} |V(k)|$ is still proportional to $M^n$.

Lemma 3.2 When $L$ is the $n^{th}$ order integrator, there exist two constants $0 < c_1 < c_2$ such that, for $M$ large enough,

$$\forall U \in S, \ c_1 M^n \leq \max_{1 \leq k \leq M} |V(k)| \leq c_2 M^n, \text{ where } V = L[U]$$

This is proved in appendix A.1. As a natural consequence, we have

Fact 3.3 There exists a constant $c > 0$ such that, for $M$ large enough, for any input $X_0 \in V_0$ which satisfies the stability condition,

$$\forall X \in V_0 \cap \Gamma(X_0), \ MSE(X_0, X) \leq \frac{c}{M^{2/3}}$$

Proof: Use the necessary condition (3.15), the lower bound of lemma 3.2, use the fact that $MSE(X_0, X) = \lambda^2$ and take $c = \frac{1}{\lambda ^2}$.

Of course, this upper bound is not satisfactory, since the classical reconstruction itself has a $\frac{1}{M^{2/3}}$ behavior. In particular, when $n = 0$ such as in dithered or predictive ADC, fact 3.3 does not show any expectation of MSE improvement with increasing oversampling ratio. The weakness of this upper bound comes from the fact that, in the application of (3.13), we did not do better than assuming that $d_{+}^k(k) = 1, \forall k, U$. Even if $d_{+}^k(k)$ is strongly correlated in time, we would expect $d_{+}^k(k)$ to be sometimes less than 1. Think that, in (3.13), it is the smallest $d_{+}^k(k)$ for $k = 1, ..., M$ which will impose the strongest constraint on $\lambda$. In fact, this is not entirely true because $|V(k)|$ also varies with $k$. But, still, this idea gives us a first hint to a deeper analysis. Let us see what happens if we assume that for any sequence
of signs \((e_1, ..., e_M), (d_{e_1}^0(1), ..., d_{e_M}^0(M))\) are independent random variables with a uniform distribution in \([0, 1]\). This is completely equivalent to saying that the quantization error signal \(e_0\) is white with a uniform distribution in \([-\frac{1}{2}, \frac{1}{2}]\). We know this is not theoretically true, but this approach will still give us some insight. As a first curiosity, we would like to know what would be the expected value of the minimum of \((d_{e_1}^0(1), ..., d_{e_M}^0(M))\) in terms of the number \(M\) of considered random variables. We have the following result:

**Fact 3.4** Let \((d(1), ..., d(M))\) be \(M\) independent random variables with a uniform distribution in \([0, 1]\), and \(d_{\text{min}} = \min_{1 \leq k \leq M} d(k)\). Then \(d_{\text{min}}\) is a random variable and its expected value is \(\frac{1}{M+1}\).

**Proof:** \(\text{Prob}(d_{\text{min}} \geq \lambda) = \text{Prob}(\forall k = 1, ..., M, d(k) \geq \lambda) = (1 - \lambda)^M\)
\[
\text{Prob}(d_{\text{min}} \in [\lambda, \lambda + d\lambda]) = -\frac{d}{d\lambda} (\text{Prob}(d_{\text{min}} \geq \lambda)) d\lambda = M(1 - \lambda)^{M-1} d\lambda
\]
\[
E(d_{\text{min}}) = \int_0^1 \lambda M(1 - \lambda)^{M-1} d\lambda = \frac{1}{M+1}
\]

If by chance the maximum of \(|V(k)|\) and the minimum of \(d_{e_k}^0(k)\) are achieved at the same time, then (3.13) will imply
\[
\lambda \leq \frac{1}{\max_{1 \leq k \leq M} |V(k)|} \min_{1 \leq k \leq M} d_{e_k}^0(k)
\]
which qualitatively reduces the upper bound of \(\lambda\) by a factor of \(\frac{1}{M}\), and the upper bound of \(\lambda^2\) by \(\frac{1}{M^2}\). Of course this is not likely to happen, but we will see that some compromise will be found.

One has to think of \(|V(k)|\) as a slowly varying function, and this for two reasons: \(V\) is the result of an \(n\)th order integration of some signal \(U\), which itself is a slowly varying function, especially for high oversampling ratios. We just want to give the intuition that the behavior \(|V(k)| \geq c_1 M^n\) is true at a substantial number of time indices other than when \(|V(k)|\) achieves its maximum. Actually, it will appear in our derivation that the upper bound of the MSE expectation is more precisely related to the average of \(|V(k)|\). The following lemma says that the average of \(|V(k)|\) has the same behavior as its maximum value, with respect to \(M\).

**Lemma 3.5** When \(L\) is the \(n\)th order integrator, there exist two constants \(0 < c_3 < c_4\) such that, for \(M\) large enough,
\[
\forall U \in \mathcal{S}, \quad c_3 M^n \leq \frac{1}{M} \sum_{k=1}^M |V(k)| \leq c_4 M^n, \quad \text{where} \quad V = L[U]
\]
This is proved in appendix A.1. As a consequence, we have the following fact:

**Theorem 3.6** Suppose that when the input signal \(X_0\) is taken at random in the stability range of \(V_0\), for any \(M \geq 1\), the quantization error values \((e_0(1), ..., e_0(M))\) are independent random variables with a uniform distribution in \([0, 1]\). Then there exists a constant \(c > 0\) such that, for \(M\) large enough,
\[
E(MSE(X_0, X)) \leq \frac{c}{M^{2n+2}}
\]
where \(X\) is taken at random in \(V_0 \cap \Gamma(X_0)\).
Proof: We first introduce some notations.
- $\vec{X} \parallel \vec{Y}$ means that $\vec{X}$ is parallel to $\vec{Y}$ with the same direction.
- $\text{Prob}(A / B)$ is the probability of event “$A$” given the event “$B$”.

It is understood that the expectation will be calculated on the basis of the random variable defined by the couple $(X_0, X)$. The element $X$ is correlated to $X_0$ in the sense that, $X_0$ is the first random element to be drawn, then $X$ is drawn inside $V_0 \cap \Gamma(X_0)$. The probabilistic knowledge of $X_0$ is given through the statistical assumption on the random variables $d_{ik}^0$. No assumption is made about the probability distribution of $X$ except the purely logical fact that $X \in V_0 \cap \Gamma(X_0)$. For every realization of $(X_0, X)$, we call $Y$ the signal (shown in figure 3.2) such that $(\vec{Y} - \vec{X}_0) / (\vec{X} - \vec{X}_0)$ and $Y$ belongs to the boundary of $V_0 \cap \Gamma(X_0)$. Then, $(X_0, Y)$ becomes a new random couple. It is obvious that

$$MSE(X_0, X) \leq MSE(X_0, Y) \tag{3.16}$$

Let us find an upper bound to the expectation of $MSE(X_0, Y)$. We first study the conditioned expectation $E_u(MSE(X_0, Y))$ of $MSE(X_0, Y)$ given that $Y$ is located in a fixed direction $\vec{U}$ with regard to $X_0$ (figure 3.2). We define the conditioned probability density

$$f_u(\lambda) d\lambda = \text{Prob} \left( ||\vec{Y} - \vec{X}_0|| \in [\lambda, \lambda + d\lambda] \mid (\vec{Y} - \vec{X}_0) \parallel \vec{U} \right)$$

Then

$$E_u(MSE(X_0, Y)) = \int_{\lambda=0}^{+\infty} \lambda^2 f_u(\lambda) d\lambda$$

If we define the cumulative probability

$$P_u(\lambda) = \text{Prob} \left( ||\vec{Y} - \vec{X}_0|| \geq \lambda \mid (\vec{Y} - \vec{X}_0) \parallel \vec{U} \right)$$

we will have $f_u(\lambda) = -\frac{d}{d\lambda} P_u(\lambda)$. Saying that $||\vec{Y} - \vec{X}_0|| \geq \lambda$ given that $(\vec{Y} - \vec{X}_0) \parallel \vec{U}$ is equivalent to saying that the boundary point $Y$ of $V_0 \cap \Gamma(X_0)$ located in the direction $\vec{U}$ with regard to $X_0$, is at a distance from $X_0$ greater than $\lambda$. This is just equivalent to saying that $X_0 + \lambda \vec{U} \in V_0 \cap \Gamma(X_0)$ (since, it is a convex set). Then

$$P_u(\lambda) = \text{Prob} \left( X_0 + \lambda \vec{U} \in V_0 \cap \Gamma(X_0) \right)$$

If we call $V = L[\vec{U}]$ and $e_k = \text{sign}(V(k))$ for $k = 1, \ldots, M$, then we can use equivalence (3.13). Since $\lambda$ and $\vec{U}$ are fixed, the probability that $X_0 + \lambda \vec{U} \in V_0 \cap \Gamma(X_0)$ completely depends on the behavior of the random variables $(d_{e_1}^0, \ldots, d_{e_M}^0)$. We have

$$P_u(\lambda) = \text{Prob} \left( \forall k = 1, \ldots, M, \quad \lambda |V(k)| \leq d_{e_k}^0 \right) \tag{3.17}$$

Let $k_0$ be the time index when $|V(k_0)|$ achieves its maximum. Whenever $\lambda > \frac{1}{|V(k_0)|}$, then $\lambda |V(k)| \leq d_{e_k}^0 (k_0) \leq 1$ becomes impossible. Therefore,

$$\forall \lambda > \frac{1}{|V(k_0)|} \quad P_u(\lambda) = 0 \quad \text{and} \quad f_u(\lambda) = 0$$

32
Performing an integration by part, we have
\[
E_U(\text{MSE}(X_0, Y)) = \left[ -\lambda^2 P_U(\lambda) \right]_0^{+\infty} - \int_0^{+\infty} (-2\lambda) P_U(\lambda) d\lambda = 2 \int_0^{+\infty} \lambda P_U(\lambda) d\lambda
\]
Suppose that \( 0 \leq \lambda \leq \frac{1}{\sqrt{\langle e_0 \rangle}} \). We have assumed that \((e_0(1), \ldots, e_0(M))\) are independent random variables with a uniform distribution in \([-\frac{1}{2}, \frac{1}{2}]\). Using equation (3.11), this implies that, for the fixed choice of signs \((e_1, \ldots, e_M)\), \((d_{e_1}^0(1), \ldots, d_{e_M}^0(M))\) are also independent with a uniform distribution in \([0, 1]\). Therefore, from (3.17) we have
\[
P_U(\lambda) = \prod_{k=1}^{M} (1 - \lambda|V(k)|)
\]
(3.18)
Using the inequality \(1 + x \leq e^x\) and the lower bound of lemma 3.5, we find
\[
P_U(\lambda) \leq \exp \left( -\lambda \sum_{k=1}^{M} |V(k)| \right) \leq e^{-\lambda \langle e_0 \rangle M n + 1}
\]
(3.19)
Actually, this inequality is true for every \( \lambda \geq 0 \); when \( \lambda > \frac{1}{\sqrt{\langle e_0 \rangle}} \), (3.19) is trivially satisfied since \( P_U(\lambda) = 0 \) and the right hand side is always positive. Therefore
\[
E_U(\text{MSE}(X_0, Y)) \leq 2 \int_0^{+\infty} \lambda e^{-\lambda \langle e_0 \rangle M n + 1} d\lambda = \frac{2}{\langle e_0 \rangle^2} \frac{1}{M^{2n+2}}
\]
The last term has been obtained by an integration by part. Thanks to lemma 3.5, we have managed to bound the expectation of \( \text{MSE}(X_0, Y) \) conditioned on \( U \in S \), independently of \( U \). This shows that, for \( M \) large enough
\[
E(\text{MSE}(X_0, X)) \leq E(\text{MSE}(X_0, Y)) \leq \frac{2}{\langle e_0 \rangle^2} \frac{1}{M^{2n+2}} \quad \square
\]
Even if the assumption of a “white” quantization noise is not theoretically justified, we still considered it for two reasons. First, starting from the “white noise” assumption usually adopted in oversampled A/D conversion, we have just shown that there is actually more information contained in the code sequence \( C_0 \) about the source signal \( X_0 \) than expected. We will see in chapter 5 that the practical results we obtained from numerical tests agree with our analytical evaluation based on this noise model. Second, the relatively simple equations we obtained can be the starting point to more detailed discussions. The assumption of quantization error independence and uniformity is exactly used when deriving equation (3.18) from (3.17). In fact, we don’t really need (3.18) to be an equality, since we only try to find an upper bound to \( P_U(\lambda) \). To obtain the \( \frac{1}{M^{2n+2}} \) behavior, it would be sufficient to have, in place of (3.18)
\[
P_U(\lambda) \leq c_0 \prod_{k=1}^{M} (1 - \lambda|V(k)|)
\]
(3.20)
where \( c_0 \) is a constant independent of \( M \). Now, \( P_U(\lambda) \), given by (3.17), is the probability that the \( M \) dimensional variable
\[
\vec{d} = (d_{e_1}^0(1), \ldots, d_{e_M}^0(M)) \quad \text{in} \quad [0, 1] \times \cdots \times [0, 1]
\]
belongs to the parallelepiped
\[ [\lambda |\nu(1)|, 1] \times \cdots \times [\lambda |\nu(M)|, 1] \] (3.21)

Inequality (3.20) just means that the ratio between the probability that \( \tilde{d}^0 \) belongs to a parallelepiped of type (3.21) and the volume of this parallelepiped, is bounded regardless of \( M \). This does not necessarily requires that \( \tilde{d}^0 \) have a probability density equal to 1 on \([0, 1] \times \cdots \times [0, 1]\) (independence and uniformity of \( d^0(k) \)). Since we are only interested in the probability of \( \tilde{d}^0 \) calculated over volumes of type (3.21) which are finite, we can even allow \( \tilde{d}^0 \) to have a Dirac type of distribution. This is theoretically the case of \( \Sigma \Delta \) modulation for example. Indeed, \( \tilde{d}^0 \) is locally an affine function of \( X_0 \) which itself is confine into a space of dimension \( 2N + 1 < M \).

There is however a known case where the \( \frac{1}{M^2} \) behavior fails. Hein and Zakhor [9, 11] proved that, in the case of 1st order \( \Sigma \Delta \) modulation with constant inputs, the MSE of reconstruction is lower bounded by \( \frac{1}{M^2} \) where \( \alpha \) is a constant. However, we found numerically (chapter 5) that a \( \frac{1}{M^2} \) behavior is recovered as soon as inputs become sinusoidal with at least an amplitude of 1/2000.

### 3.4 \( n^{th}\) order multi-quantizer converter

We presented in section 2.5 (figure 2.11a) the general structure of what we called multi-quantizer converter. When considering a fixed analog input \( X_0 \) giving \( C_0 \), we saw that \( \Gamma(X_0) = \Gamma(C_0) \) is equal to the intersection of the convex sets \( \Gamma_i(X_0) = \Gamma_i(C_0) \) which is the signal cell corresponding to the \( i^{th} \) single-quantizer converter of the virtual structure (figure 2.13). Therefore, we have

\[
V_0 \cap \Gamma(X_0) = (V_0 \cap \Gamma_1(X_0)) \cap (V_0 \cap \Gamma_2(X_0)) \cap \ldots \cap (V_0 \cap \Gamma_p(X_0))
\]

suppose that the order of the \( i^{th} \) single-quantizer converter in the original structure is \( n_i \). Using formula (2.1) in section 2.5, the order of \( H^i \) will be

\[
n_i' = n_i + n_{i-1} + \cdots + n_1
\]

If we apply the result of theorem 3.6, the expected value of \( \text{MSE}(X_0, X) \) when \( X \in V_0 \cap \Gamma(X_0) \) will be of the order of \( O(M^{-2n'_p+2}) \). Since \( n'_p = \sum_{i=1}^{p} n_i \) is the highest order, \( V_0 \cap \Gamma_p(X_0) \) will be qualitatively speaking the smallest set containing \( X_0 \). We conclude that the "size" of \( V_0 \cap \Gamma(X_0) \) in terms of MSE has the order \( O(M^{-2n+2}) \) where \( n = n'_p \). In practice, while looking for an estimate in \( V_0 \cap \Gamma(X_0) \), we will only work with \( V_0 \cap \Gamma_p(X_0) \). This means we have reduced the study of our multi-quantizer converter to the last single-quantizer converter in the virtual structure. This conclusion should not be misleading. Looking at figure 2.13, it seems as if we dropped the use of the other code components \( C_{01}, C_{02}, \ldots, C_{0(p-1)} \). But one has to remember that this code information is actually included in the dither sequence \( D_{0p} \). We have shown in section 2.5 (figure 2.14e) the relation between \( D_{0p} \) and \( C_0, \ldots, C_p \) in the case \( p = 2 \). Therefore, the definition of the last single-quantizer converter of the virtual structure indeed contains the full information of the output code sequence \( C_0 = (C_{01}, \ldots, C_{0p}) \).
3.5 Conclusion

Working on periodic input signals, we have shown under certain assumptions that picking an estimate $X$ of an original signals $X_0$, having the same bandwidth and producing the same digital sequence, leads to an MSE upper bounded by $R^{-2(n+2)}$ instead of $R^{-2(n+1)}$, where $R$ is the oversampling rate and $n$ is the order of the converter. For simple ADC ($n = 0$), this is conditioned on the fact that the original signal $X_0$ has a minimum number of threshold crossings. For single-quantizer converters of higher order ($n \geq 1$), the MSE has a necessary upper bound proportional to $R^{-2(n+2)}$ when assuming no-overload stability and a white and uniform quantization error signal. We explained, however, that this result can be obtained with weaker assumptions. This analysis can be applied to a multi-quantizer converter since it can be decomposed into single-quantizer converters working in parallel, according to chapter 2. As a consequence, the overall performance is still $R^{-2(n+2)}$ where $n$ is the total order of the multi-quantizer converter.
Chapter 4

Algorithms for the projection on \( \Gamma_I(C_0) \)

4.1 Introduction

In this chapter, we propose computer implementable algorithms to perform the projection of a signal \( X \) on \( \Gamma_I(C_0) \) for a given single-quantizer converter, a code sequence \( C_0 \) and a time index subset \( I \). The set \( \Gamma_I(C_0) \) is defined in section 2.4 and is systematically derived from the virtual converter. When \( I \) is the whole time range, then \( \Gamma_I(C_0) \) coincides with \( \Gamma(C_0) \), the set of all analog signals giving code sequence \( C_0 \) through the original converter. We saw that \( \Gamma_I(C_0) \) is always convex. Even if our study is based on the general case where \( I \) is not necessarily the full time range, whenever we can, we will try to find a method to perform the projection on \( \Gamma(C_0) \), since \( \Gamma(C_0) \) gives the best localization of \( X_0 \) based on the knowledge of \( C_0 \). Otherwise, we will try to find the projection on \( \Gamma_I(C_0) \) where \( I \) is as large as possible, or, in other words, \( \Gamma_I(C_0) \) is as close as possible to \( \Gamma(C_0) \). Our particular motivation will be to test the performance of the proposed algorithm in the context of oversampled ADC (chapter 5), and compare the results with the analysis of chapter 3.

From section 2.3 we know that the projection of \( X \) on \( \Gamma_I(C_0) \) is the signal \( X' \in \Gamma_I(C_0) \) which minimizes the mean squared error with respect to \( X \). We immediately recognize the problem on a constrained minimization: \( X' \) minimizes the function \( Y \mapsto MSE(X,Y) \) subject to the constraint \( Y \in \Gamma_I(C_0) \). We can then apply the theory of optimization in the context of quadratic minimization function with convex constraints. However, we have to take into account the fact that elements we want to optimize are signals defined on a whole time interval. It is of course expected that a computer can only process a signal through a moving finite length window (typically, the length of the buffer). Conceptually, the total length of signals should be thought of as infinite compared to that of the window of operation. Therefore, we have to design specific optimization algorithms, adapted to this practical constraint. We will see in section 4.3 that, in the case of zero\(^{th}\) order conversion (simple, dithered or predictive ADC), the minimization of \( MSE(X,X') \) subject to \( X' \in \Gamma_I(C_0) \) can be performed by an independent computation of \( X'(k) \) at time \( k \), only function of \( X(k) \) and \( D_0(k) \) at the same instant (\( D_0 \) is the dither sequence derived from the virtual converter of figure 2.9). This will enable us to propose a straightforward algorithm for the projection on \( \Gamma(C_0) \). Unfortunately, this is no longer true with converters of higher order. For 1\(^{st}\) order \( \Sigma\Delta \) modulation (section 4.4) we will see that \( X'(k) \) has to be computed by blocks in time. However, the computation is independent from one block to another. We propose an
algorithm which finds the projection on $\Gamma(C_0)$ which we call the “thread algorithm”.

With converters of order higher than 1, it becomes no longer possible to split the computation of $X'(k)$ into independent time intervals. Indeed, the determination of $X'(k)$ at a certain instant $k$ will depend on the knowledge of $C_0$ and $X$ on the whole time range. Even if we assume that the computer's buffer is longer than the signal, there is no single step method to find the minimum to the MSE subject to inequality constraints (the relation $X' \in \Gamma_I(C_0)$ can indeed be formulated as inequality constraints). Based on a property (conjecture 4.6) we proved for the case $n = 2$, and which still works for higher orders as confirmed by our numerical simulations, we propose an algorithm which finds in one step the projection on $\Gamma_I(C_0)$, where $I$ is recursively constructed in time. If we assume that the computer's buffer is longer than the signal, this algorithm gives the exact projection on $\Gamma_I(C_0)$. When the buffer is limited, we show qualitatively that the solution given by the algorithm is a “satisfactory” approximation of the projection.

4.2 Specific formulation of the constrained minimization problem and properties

In the formulation of the constrained minimization problem, is is particularly convenient to take $X$, the signal we want to project, as the origin of the signal space. This is possible since $X$ is always known. We will then express any element of $V$ in the form $X + x$. The projection of $X$ on $\Gamma_I(C_0)$ becomes the signal $X + x \in \Gamma_I(C_0)$ such that $\|x\|_V$ is minimized. The function to minimize is simple but the system of constraints $\Gamma_I(C_0)$ has a complex expression. Therefore, we propose to perform a change of variable on $x$ which will greatly simplify the expression of constraints while the minimization function will be of a quadratic type.

It is possible to determine the signal $A$ seen by the quantizer when $X$ is fed into the virtual converter (figure 4.1). When any signal $X + x$ is input, the signal seen by the quantizer is of the form $A + a$, where $a = L[x]$ and $L$ is the linear part of $H$. By assumption, $L$ is invertible and defines a one-to-one mapping between $x$ and $a$. According to the structure of the virtual converter, $X + x$ belongs to $\Gamma_I(C_0)$ if and only if, for every time index $k \in I$, $A(k) + a(k)$ belongs to the quantization interval labeled $C_0(k)$, or equivalently, $a(k)$ belongs to this interval shifted by $-A(k)$. Let us introduce the following notation: for every $k$ (not necessarily in $I$), we call $Q_{c_k, X}(k)$ the quantization interval labeled $C_0(k)$ shifted by $-A(k)$. For a fixed $k$, $Q_{c_k, X}(k)$ does only depend on $C_0$ and $X$: the identification of the
quantization interval is directly related to \( C_0(k) \), and signal \( A \) depends on \( X \) and \( D_0 \), where \( D_0 \) itself depends on \( C_0 \). With the following equivalence, we express the constraints in terms of variable \( a \):

\[
X + x \in \Gamma_I(C_0) \iff a = L[x] \in C_{c_0,X}(I)
\]

where

\[
C_{c_0,X}(I) = \left\{ a \in \mathcal{V} \mid \forall k \in I, a(k) \in Q_{c_0,X}(k) \right\}
\]

In the following, since \( C_0 \) and \( X \) are considered to be fixed, we will simply write

\[
Q_{c_0,X}(k) = Q(k) \quad \text{and} \quad C_{c_0,X}(I) = C(I)
\]

When \( I \) covers the whole time range, we will just write \( C(I) = C \). Let us just keep in mind the relation of \( Q(k) \) and \( C(I) \) with \( C_0 \) and \( X \). Figure 4.2 shows a time representation of \( C(I) \).

The function to minimize \( \|x\|_\mathcal{V} \) can also be expressed in terms of \( a \). We define

\[
\phi(a) = \frac{1}{2} \left\| L^{-1}[a] \right\|_\mathcal{V}^2
\]

If \( a = L[x] \), then \( \phi(a) = \frac{1}{2} \|x\|_\mathcal{V}^2 \).

We can now express the projection problem as follows:

\footnote{Again, we will always consider the closure of \( C_{c_0,X}(I) \). This is equivalent to considering the closure of every \( Q_{c_0,X}(k) \). As a consequence, we will assume that \( Q_{c_0,X}(k) \) includes its bounds.}
**Fact 4.1** The projection of $X$ on $\Gamma_f(C_0)$ is the signal $X + x$ such that $x = L^{-1}[a]$ and $a$ is the minimum of $\phi$ subject to the constraint $C(I) = \{a \in V/ \forall k \in I, a(k) \in Q(k)\}$

Since $\phi$ is a quadratic function and $C(I)$ a convex set, the theory of optimization directly confirms that $\phi$ has a unique minimum subject to $C(I)$. This theory also provides a characterization of the minimum when $\phi$ is quadratic, which is the case here. If we designate by $\frac{\partial \phi}{\partial a_0}$ the sequence whose value $\frac{\partial \phi}{\partial a_0}(k)$ at time $k$ is the partial derivative of $\phi(a)$ with respect to $a(k)$, we have the following theorem:

**Theorem 4.2 (Euler inequality characterization)** A signal $a_0$ is a minimum of $\phi$ subject to convex constraints $S$ if and only if

$$a_0 \in S \quad \text{and} \quad \forall a \in S, \quad \sum_k \frac{\partial \phi}{\partial a_0}(k) \cdot (a(k) - a_0(k)) \geq 0 \quad (4.1)$$

Thanks to the change of variable $x \leftrightarrow a$, the constraint specification $C(I)$ has the particular advantage to be separable. By this we mean that constraints are defined independently of $a(k)$ at every time index $k$. As a consequence of Euler characterization, the theory of optimization provides us with the following theorem:

**Theorem 4.3** Let $a_0$ be a minimum of $\phi$ subject to separable convex constraints $S$. Then

$$\forall k, \forall a \in S, \quad \frac{\partial \phi}{\partial a_0}(k) \cdot (a(k) - a_0(k)) \geq 0 \quad (4.2)$$

*If for a certain index $k$, $S$ does not impose any constraint on $a(k)$, then*

$$\frac{\partial \phi}{\partial a_0}(k) = 0 \quad (4.3)$$

Proof: Let us consider a fixed time index $k_1$, and a fixed element $a_1 \in S$. Let $a'_1$ be the sequence equal to $a_0$ everywhere except at instant $k = k_1$ where $a'_1(k_1) = a_1(k_1)$. Since $a_0 \in S$ and $S$ is separable, $a'_1$ also belongs to $S$. Applying (4.1) on $a = a_1$, we find $\frac{\partial \phi}{\partial a_0}(k_1) \cdot (a_1(k_1) - a_0(k_1)) \geq 0$. This proves (4.2). To show (4.3), suppose that at $k = k_1$, $S$ does not impose any constraint on $a(k_1)$. Let $a_+, a_-$ be the sequences equal to $a_0$ everywhere except at $k = k_1$ where $a_+(k_1) = a_0(k_1) + 1$ and $a_-(k_1) = a_0(k_1) - 1$. We have $a_+, a_- \in S$. Then applying (4.1) on $a = a_+$ and $a = a_-$, we respectively find $\frac{\partial \phi}{\partial a_0}(k_1) \geq 0$ and $-\frac{\partial \phi}{\partial a_0}(k_1) \geq 0$, which implies that $\frac{\partial \phi}{\partial a_0}(k_1) = 0$.

If the change of variable $x \leftrightarrow a$ simplifies the constraints expression, it however complicates the function to minimize $\phi$. In this chapter, we will only consider converters where $L$ is an $n$th order integrator. In this case, using notation 4.4, $\frac{\partial \phi}{\partial a_0}$ can be obtained according to fact 4.5.

**Notation 4.4** $n$th order discrete derivative

For $y \in V$, $y^{(n)-}$ and $y^{(n)+}$ are the $n$th order discrete backward and forward derivatives of $y$ recursively defined as follows:

$$y^{(0)-} = y^{(0)+} = y$$

39
and for $n \geq 0$

\[
\begin{align*}
y^{(n+1)}(1) &= y^{(n)}(1) \\
y^{(n+1)}(k) &= y^{(n)}(k) - y^{(n)}(k-1), \text{ for } k \geq 1
\end{align*}
\]

and

\[
y^{(n+1)}(k) = y^{(n)}(k) + y^{(n)}(k+1), \text{ for } k \geq 0
\]

**Fact 4.5** If $L$ is an $n^{th}$ order integrator (figure 2.7) and $a = L[x]$, then

\[
x = a^{(n)} \quad \text{and} \quad \frac{\partial \phi}{\partial a} = (-1)^n x^{(n)}
\]

This is shown in appendix A.2.

### 4.3 Zeroth order converter (simple, dithered, predictive ADC)

This is the easy case where $H = L = \text{identity}$. We will immediately consider the projection on $\Gamma(C_0)$. We will derive an algorithm directly without using any of the theorems of the previous section.

Signal $A$ is just $X - D_0$ where $D_0$ is the zero signal in simple ADC, the dither signal in dithered ADC and the feedback output in predictive ADC. Variables $x$ and $a$ are thus identical and the function to minimize is $\phi(a) = \frac{1}{2}||a||^2$ which is proportional to $\sum_k |a(k)|^2$. Since the constraint $C(I)$ is a system of conditions applied to samples $a(k)$ taken separately, the minimization of $\sum_k |a(k)|^2$ is equivalent to the individual minimization of each $|a(k)|$. If $k \in I$, $a_0(k)$ is necessarily the element of $Q(k)$ of lowest magnitude. Figure 4.3 shows an example of minimization solution. This leads to the following algorithm:

**Algorithm 1**: Given the knowledge of $C_0$ and $X$, at every time index $k$:

1. calculate $D_0(k)$ the value of the feedback signal $G[C_0]$ at time $k$
2. calculate $A(k) = X(k) - D_0(k)$
3. determine $Q(k)$, the quantization interval labeled $C_0(k)$ shifted by $-A(k)$
4. if $Q(k)$ contains 0, take $a_0(k) = 0$
   otherwise, take $a_0(k)$ equal to the closest bound of $Q(k)$ to 0.

Then, $X + a_0$, where $a_0$ results from this algorithm, is the projection of $X$ on $\Gamma_I(C_0)$.

In the particular case of simple ADC ($D_0 = 0$), one can check that this leads to the transformation we introduced in chapter 1 (figure 1.2).

### 4.4 1st order converter (1st order $\Sigma\Delta$ modulation)

In this case $L$ is a single discrete integrator ($n = 1$). According to fact 4.5, if $L[x] = a$, then $x(k) = a(k) - a(k-1)$ is the slope of $a$ at time $k$ and

\[
\frac{\partial \phi}{\partial a} = -\{x(k + 1) - x(k)\} = -\{(a(k + 1) - a(k)) - (a(k) - a(k - 1))\}
\]

is minus the change of slope of $a$ about time index $k$. Here again, we consider that $I$ covers the whole time range. We propose an algorithm that we describe in a “physical” way.
Algorithm 2: Given the knowledge of code sequence $C_0$ and signal $X$,
1. calculate signal $A = H[X] - G[C_0]$
2. determine, for every $k$, $Q(k)$ the quantization interval labeled $C_0(k)$ shifted by $-A(k)$
3. represent $C$ in time (like in figure 4.2)
4. attach a “thread” at node $(k = 0, a(0) = 0)$, stretch it under tension between the constraints (arrows in figure 4.2)
5. take $(a_0(k))_{k>0}$ on the path of the resulting thread position.

The solution $a_0$ resulting from this algorithm is shown in figure 4.4. Is is easy to see that this signal satisfies criteria (4.1) of theorem 4.2. First, it is obvious that $a_0 \in C$. Whenever $\frac{\partial \phi}{\partial a}$ is non zero, the thread touches a constraint (in figure 4.4, contact points $k = 5, 9, 12, 14, ...$). But the way the thread’s slope changes is related to the direction of the arrow of contact. For example, in figure 4.4, the change of slope is negative only when the arrow of contact goes up ($k = 5, 9$) and vice-versa ($k = 12$). When an arrow of contact goes up, for example at $k = 5$, for any other admissible signal $a \in C$, then we necessarily have $a(k) - a_0(k) \geq 0$. Similarly, if an arrow of contact goes down at $k = 12$, then $a(k) - a_0(k) \leq 0$. Since $\frac{\partial \phi}{\partial a}$ is the opposite to the slope’s variation, we have just shown that whenever $\frac{\partial \phi}{\partial a_0}(k) \neq 0$, then $\frac{\partial \phi}{\partial a_0}(k)$ and $a(k) - a_0(k)$ have the same sign. This implies (4.1). Therefore $a_0$ is the minimum of $\phi$ subject to $C$. Then, once $a_0$ is computed, the projection of $X$ on $\Gamma_{I}(C_0)$ is the signal $X + x_0$ where $x_0 = \frac{1}{L} a_0 = a_0^{(1)}$ is just the slope of $a_0$, or the slope of the thread in figure 4.4. This algorithm is easy to translate in terms of computer operations.
Figure 4.4: Solution \( a_0 \) for 1\(^{st} \) order \( \Sigma \Delta \) modulation. This is obtained from the “thread algorithm”.

### 4.5 Higher order converter (\( n^{th} \) order \( \Sigma \Delta \) modulation)

#### 4.5.1 Introduction

The principle of the “thread algorithm” was to find the sequence which meets the constraints and, at the same time, minimizes its “slope”. For an \( n^{th} \) order converter, the principle is similar, but the function to minimize is the energy of the \( n^{th} \) order derivative. For example, when \( n = 2 \), we will have to minimize the “curvature” of the signal. This is similar to the physical problem of thin elastic beam under constraint. Figure 4.5 shows the solution minimizing the curvature under the same constraints as in figure 4.4.

However, the resolution of this problem includes an extra difficulty. Looking at first order minimization, we can see that the solution is obtained by linear interpolation of two consecutive contact points. The thread algorithm gave an easy way to identify these contact points. With 2\(^{nd} \) order minimization, the way the solution is interpolated between two contact points depends on the past but also on how the signal will be constrained on the future. It is not even possible to determine locally whether a constraint will be a contact point of the final solution or not.

Therefore, we will no longer try to compute the projection on \( \Gamma(C_0) \), but only on \( \Gamma_I(C_0) \), where we will try to maximize the time density of \( I \). First, we will assume that the computer has a buffer as long as signals to be processed. In this context, we will present an algorithm which, at every time index \( k \), proposes a choice \( I_p \) of \( p \) time indices belonging to the past from \( k = 0 \), and compute the minimum \( a_p \) of \( \phi \) subject to \( C(I_p) \) from the initial time index 0 to the present time index \( k \). The construction of \( I_p \) will be such that \( I_p = I_{p-1} \cup \{k_p\} \).
where $k_p$ belongs to the future of $I_{p-1}$. The computation of $a_p$ will be obtained by correcting $a_{p-1}(k)$ from $k = 0$ to $k = k_{p-1}$, previously computed and stored in a buffer, and adding to the same buffer the specific values of $a_p(k)$ computed from $k = k_{p-1} + 1$ to $k + k_p$. This algorithm will provide the exact projection on $\Gamma_f(C_0)$ where $I = I_{p_m}$ and $p_m$ is the total number of selected time indices. If the computer has a buffer limited to length $l$, we will use the fact that, when computing $a_p$ from $a_{p-1}$, the correction signal $a_p(k) - a_{p-1}(k)$ decays exponentially when $k$ goes from $k_{p-1}$ to the past. We will then limit ourselves to compute $a_p$ by a correction of the stored value of $a_{p-1}(k)$ from $k = k_{p-1} - l + 1$ to $k = k_{p-1}$. This will lead to an approximation of the projection on $\Gamma_f(C_0)$.

### 4.5.2 Recursion principle and first algorithm

Let us consider a sequence $(k_i)_{i \geq 1}$ of increasing time indices ($\forall i \geq 1$, $0 < k_i < k_{i-1}$). We call $I_p = \{k_1, \ldots, k_p\}$ and $a_p$ the minimum of $\phi$ subject to $C(I_p)$. We introduce the minimum $b_p$ of $\phi$ subject to the equality constraints

$$E(I_p) = \{a \in \mathcal{V} / \forall i = 1, \ldots, p-1, a(k_i) = 0 \text{ and } a(k_p) = 1\}$$

**Conjecture 4.6** If $(k_i)_{i \geq 1}$ is such that $\forall i \geq 1$, $k_i - k_{i-1} \geq n$ ($k_0 = 0$ by convention), then for $p \geq 1$, $b_p$ verifies

$$\forall i = 1, \ldots, p, \quad \text{sign} \left( \frac{\partial \phi}{\partial b_p}(k_i) \right) = (-1)^{p-i}$$

(4.4)

We discovered this property numerically when $L$ is an $n^{th}$ order integrator. We only have to prove in the case $n = 2$ (see appendix A.6.2). As a consequence, we have the following fact:
**Fact 4.7** We suppose that \( \forall i \geq 1, \ k_i - k_{i-1} \geq n \) \((k_0 = 0)\). If for a certain \( p \geq 2 \), \( a_{p-1} \) verifies

1. \( \forall i = 1, ..., p-1 \), \( \text{sign} \left( \frac{\partial \phi}{\partial a_{p-1}}(k_i) \right) = \epsilon(-1)^{p-i} \ \text{where} \ \epsilon \in \{-1, 1\} \)
2. \( a_{p-1}(k_p) \) is below (resp. above) interval \( Q(k_p) \) when \( \epsilon = 1 \) (resp. -1)

then \( a_p \) verifies

3. \( \forall i = 1, ..., p \), \( \text{sign} \left( \frac{\partial \phi}{\partial a_p}(k_i) \right) = \epsilon(-1)^{p-i} \)
4. \( a_p = a_{p-1} + \epsilon \delta b_p \), where \( \delta \) is the distance between \( a_{p-1}(k_p) \) and \( Q(k_p) \)

This is a consequence of (4.4), the Euler characterization, theorem 4.3 and the linearity of \( a \mapsto \frac{\partial \phi}{\partial a} \) (fact 4.5). The complete proof is shown in appendix A.3. This fact is still true at \( p = 1 \), when taking \( a_0 = 0 \) (zero signal). Of course the criteria (1) disappears because \( I_0 = \emptyset \). Since \( \epsilon \) has not been determined yet, we will have the freedom to choose \( k_1 \) such that 0 is either below or above \( Q(k_1) \). Then \( \epsilon \) will be fixed to +1 or -1 respectively. Fact 4.7 becomes:

**Fact 4.8** Assuming that \( k_1 \geq n \), if we have

1. \( 0 \) does not belong to \( Q(k_1) \)

then \( a_1 \) verifies

3. \( \text{sign} \left( \frac{\partial \phi}{\partial a_p}(k_i) \right) = \epsilon \), where \( \epsilon = 1 \) (resp. -1) if 0 is below (resp. above) \( Q(k_1) \)
4. \( a_1 = \epsilon b_1 \), where \( \delta \) is the distance of 0 to \( Q(k_1) \)

Proof : Like in the proof of fact 4.7 (A.3) one has to see that \( \epsilon b_1 \in C(I) \) and verifies the Euler inequality with regard to \( C(I) \)

These facts give us the foundation for our next algorithm. If we are able to find a set of \( p-1 \) increasing time indices \( I_{p-1} = \{k_1, ..., k_{p-1}\} \) such that the minimum \( a_{p-1} \) of \( \phi \) subject to \( C(I_{p-1}) \) can be obtained and verifies criteria (1) of fact 4.7, then we will look for an index \( k_p \geq k_{p-1} + n \) such that criteria (2) is satisfied. Taking \( I_p = I_{p-1} \cap \{k_p\} \), if we can calculate the minimum \( b_p \) of \( \phi \) subject to \( E(I_p) \), we can calculate the signal \( a_{p-1} + \epsilon \delta b_p \) (where \( \epsilon \) and \( \delta \) are defined in criteria (1) and (4)) which is for sure the minimum \( a_p \) of \( \phi \) subject to \( C(I_p) \). According to fact 4.7, \( a_p \) verifies criteria (3) which can be written:

\[
\forall i = 1, ..., p, \ \text{sign} \left( \frac{\partial \phi}{\partial a_p}(k_i) \right) = (-\epsilon)(-1)^{p+1-i}
\]

and which is similar to criteria (1) when replacing \( p \) by \( p + 1 \) and \( \epsilon \) by \( -\epsilon \). At the first step \( p = 1 \), the initial \( \epsilon \) is determined according to the choice of \( k_1 \) and criteria (3'). We formally describe the algorithm as follows:

**Algorithm 3.1** : Given the knowledge of code sequence \( C_0 \) and signal \( X \),

1. Initialize \( p = 0 \), \( k_0 = 0 \), \( I_0 = \emptyset \), \( a_0 = 0 \) (zero signal)
2. Increment \( p \)
3. For increasing \( k \) starting from \( k_{p-1} + n \), determine \( Q(k) \) using the knowledge of \( C_0 \) and \( X \) (like in algorithm 2)
4. Choose \( k_p \geq k_{p-1} + n \) such that \( a_{p-1}(k_p) \) is below (resp. above) \( Q(k_p) \) if \( \epsilon_0(-1)^{p} = 1 \) (resp. -1) \( (\epsilon_0 \) is chosen at the same time as \( k_1 \) during the first step \( p = 1 \))
5. Calculate \( \delta_p \) the distance between \( a_{p-1}(k_p) \) and \( Q(k_p) \)
6. Take \( I_p = I_{p-1} \cap \{k_p\} \)
7. Calculate the minimum \( b_p \) of \( \phi \) subject to \( E(I_p) \)
(8) - Calculate \( a_p = a_{p-1} + \epsilon_0 (-1)^p \delta_p b_p \)

(9) - Go to (2)

An example of construction of \( I_p, a_p, b_p \) according to this algorithm is shown in figure 4.6.

### 4.5.3 More practical version for algorithm 3.1

One should not forget that the computation of \( a_p \) is only an intermediate step in the computation of the projection \( X + x_p \) of \( X \) on \( \Gamma_p(C_0) \); we still have to perform the operation \( x_p = L^{-1} [a_p] = a_p^{(n)-} \). If we apply the operator \( L^{-1} \) on the recursive relation of line (8) in algorithm 3.1, we find

\[
x_p = x_{p-1} + \epsilon_0 (-1)^p \delta_p y_p
\]

where \( y_p = L^{-1} [b_p] = b_p^{(n)-} \). However, line (4) of algorithm 3.1 still requires \( a_{p-1} (k) \) to be known for \( k > k_{p-1} \). Therefore, we will still need to determine \( b_{p-1} (k) \) for \( k > k_p \). We propose the modified algorithm:

**Algorithm 3.2**: In algorithm 3.1, replace lines (7) and (8) by

(7) - Call \( b_p \) the minimum of \( \phi \) subject to \( C(I_p) \)

- calculate \( y_p = b_p^{(n)-} \)
- calculate \( b_p (k) \) for \( k > k_p \)

(8) - calculate \( x_p = x_{p-1} + \epsilon_0 (-1)^p \delta_p y_p \)

- calculate \( a_p (k) = a_{p-1} (k) + \epsilon_0 (-1)^p \delta_p b_p (k) \) for \( k > k_p \).

This algorithm requires \( x_p \) to be memorized at every step \( p \). However, since \( x_p \) is only an intermediate variable in the computation of the final solution \( x_{p_m} \), it is not necessary to determine \( x_p \) explicitly on its full time range.

Actually, \( x_p \) is completely determined from the knowledge of

\[
x_p (1), x_p^{(1)+} (1), ..., x_p^{(n-1)+}, \quad \text{and} \quad x_p^{(n)+} (k_i) \quad \text{for} \quad i = 1, ..., p
\]

Indeed, as a consequence of theorem 4.3 and fact 4.5,

\[
\forall k \notin I_p, \quad x_p^{(n)+} (k) = 0
\]

This implies that \( x_p^{(n)+} \) is known on the full time range and \( x_p \) can be obtained from an \( n^{th} \) order integration of \( x_p^{(n)+} \). But, when reaching the final step \( p = p_m \), \( x_{p_m} \) itself has to be given explicitly; it is not realistic to compute it by an \( n^{th} \) order integration from initial conditions given at \( k = 1 \) because of round off error accumulation. Therefore, we propose an intermediate solution which consists in calculating the vectors

\[
W_{p,i} = \begin{bmatrix}
x_p^{(n-2)+} (k_i + 1) \\
\vdots \\
x_p^{(0)+} (k_i + 1)
\end{bmatrix}
\]

for every \( i = 0, ..., p \), to characterize \( x_p \). Indeed, \( x_p \) can be reconstructed from \( (W_{p,i})_{0 \leq i \leq p-1} \) using the following facts:
Figure 4.6: Example of construction of $I_p, \ a_p, \ b_p$ (with $1 \leq p \leq 3$). The selected constraints are shown by dark arrows.
Fact 4.9 \( \forall k \geq k_p + 1, \forall q \geq 0, \quad x_p^{(q)+}(k) = 0 \) and \( y_p^{(q)+}(k) = 0 \)

Proof: Suppose there exists \( h > k_p \) such that \( x_p(h) \neq 0 \). Let \( x_p' \) be the sequence equal to \( x_p \) everywhere except at \( k = h \) where \( x_p'(h) = 0 \) and call \( a_p' = L[x_p'] \). We have \( ||x_p'||_v < ||x_p||_v \) which implies \( \phi(a_p') < \phi(a_p) \). Since \( L \) is causal, \( a_p' \) will differ from \( a_p \) only for \( k \geq h \). Therefore \( a_p' \in \mathcal{C}(I_p) \) and \( \phi(a_p') \geq \phi(a_p) \) which is impossible. We conclude that \( \forall k > k_p, x_p(k) = 0 \). It will be also the case for its successive forward derivatives for \( k > k_p \). The reasoning is the same for \( y_p \). □

Fact 4.10 \( x_p^{(n-2)+} \) and \( y_p^{(n-2)+} \) are piecewise linear on intervals \( \{k_{i-1} + 1, \ldots, k_i + 1\} \) (1 ≤ \( i \leq p, k_0 = 0 \)).

For 1 ≤ \( i \leq p \), \( \forall k = k_{i-1} + 1, \ldots, k_i - 1 \), \( x_p^{(n)+}(k) = 0 \), because of theorem 4.3 and fact 4.5. Since \( x_p^{(n)+}(k) = (x_p^{(n-2)+}(k + 2) - x_p^{(n-2)+}(k + 1)) - (x_p^{(n-2)+}(k + 1) - x_p^{(n-2)+}(k)) \), by definition of the forward derivative, we conclude that

\[
\forall k = k_{i-1} + 1, \ldots, k_i - 1, \quad (x_p^{(n-2)+}(k + 2) - x_p^{(n-2)+}(k + 1)) = (x_p^{(n-2)+}(k + 1) - x_p^{(n-2)+}(k))
\]

Therefore, \( x_p^{(n-2)+} \) is piecewise linear on intervals \( \{k_{i-1} + 1, \ldots, k_i + 1\} \). The reasoning is similar for \( y_p \) □

On every interval \( \{k_{i-1} + 1, \ldots, k_i + 1\} \), \( x_p^{(n-2)+}(k) \) can be obtained by linear interpolation of the top components of \( W_{p,i-1} \) and \( W_{p,i} \). Then, \( x_p(k) \) can be obtained by an \( (n - 2) \)th order integration of \( x_p^{(n-2)+} \) on this interval, with initial conditions at \( k = k_{i-1} + 1 \) contained in \( W_{p,i-1} \). The length of integration will be this time limited to \( l_i = k_i - k_{i-1} \). In the particular case where \( i = p \), note that \( W_{p,p} \) is the zero vector according to fact 4.9. This fact also says that \( x_p(k) \) does not need to be calculated for \( k \geq k_p + 1 \) since it is necessarily equal to zero.

From line (4) of algorithm 3.2, we need to determine \( a_p(k) \) at least for \( k \geq k_p + n \). From fact 4.9 we have \( a_p^{(n)+}(k) = x_p(k) = 0 \) for \( k \geq k_p + 1 \). This implies that \( a_p(k) \) can be determined by an \( n \)th order integration of the zero signal on \( k \geq k_p + 1 \), provided we know the initial conditions at \( k = k_p \): \( a_p(k_p), a_p^{(1)+}(k_p), \ldots, a_p^{(n-1)+}(k_p) \). For this purpose, we define the following vector

\[
V_p = \begin{bmatrix}
a_p^{(n-1)+}(k_p) \\
\vdots \\
a_p^{(1)+}(k_p)
\end{bmatrix}
\] (4.6)

We will call \( W_{p,i}, V_p \) the vectors associated with \( a_p \). They all have size \( (n - 1) \). Note that in the case \( n = 2 \), they are just scalars. Similarly, let us define \( W_{p,i}^0, V_p^0 \) the vectors associated with \( b_p \) as

\[
W_{p,i}^0 = \begin{bmatrix}
y_p^{(n-2)+}(k_i + 1) \\
\vdots \\
y_p^{(0)+}(k_i + 1)
\end{bmatrix} \quad (0 \leq i \leq p) \quad \text{and} \quad V_p^0 = \begin{bmatrix}
b_p^{(n-1)+}(k_p) \\
\vdots \\
b_p^{(1)+}(k_p)
\end{bmatrix}
\] (4.7)

Algorithm 3.2 becomes
Algorithm 3.3: In algorithm 3.2, replace lines (1),(3),(4),(7) and (8) by
(1) - Initialize $p = 0$, $k_0 = 0$, $I_0 = \emptyset$, $a_0(0) = 0$, $V_0 = O_{n-1}$ (null vector of size $n - 1$)
(3) - For increasing $k$ starting from $k_{p-1} + 1$, determine
- $Q(k)$ using the knowledge of $C_0$ and $X$ (like in algorithm 2)
- $a_{p-1}(k)$ by $n^{th}$ order integration of the zero signal with initial conditions at $k = k_{p-1}$
given by $a_{p-1}(k_{p-1})$ and $V_{p-1}$
(4) - Choose $k_p \geq k_{p-1} + n$ such that $a_{p-1}(k_p)$ is below (resp. above) $Q(k_p)$ if $\epsilon_0(-1)^p = 1$
(resp. $-1$) ($\epsilon_0$ is chosen at the same time as $k_1$ during the first step $p = 1$)
Call $V'_{p-1} = \left[a_{p-1}^{(n-1)}(k_p) \cdots a_{p-1}^{(1)}(k_p)\right]^T$
(7) - Calculate $(W_{p,i})_{0 \leq i \leq p-1}$, $V^0_p$ the vectors associated with the minimum $b_p$ of $\phi$ subject
to $C(I_p)$ (see formula (4.7))
(8) - Calculate
$W_{p,i} = W_{p-1,i} + \epsilon_0(-1)^p \delta_p W^0_{p,i}$, for $0 \leq i \leq p - 2$
$W_{p,p-1} = \epsilon_0(-1)^p \delta_p W^0_{p,p-1}$
$V_p = V'_{p-1} + \epsilon_0(-1)^p \delta_p V^0_p$
$a_p(k_p) = a_{p-1}(k_p) + \epsilon_0(-1)^p \delta_p$

Line (8) requires $(W^0_{p,i})_{0 \leq i \leq p-1}$ to be memorized for the next step $(p + 1)$.

4.5.4 Computation of $W^0_{p,i}$, $V^0_p$

We show in this section that $W^0_{p,i}$ and $V^0_p$ can be calculated algebraically from the recursive
computation of a $(2n - 2) \times (n - 1)$ matrix $R_p$ and a $p(n - 1) \times (n - 1)$ matrix $M_p$. We first
introduce some notation:

Notation 4.11 For any sequence $b \in \mathcal{V}$, we define the column vector

$$U_b(k) = \left[y^{(n-2)+}(k+1) \cdots y^{(0)+}(k+1) b^{(n-1)-}(k) \cdots b^{(1)-}(k)\right]^T$$

where $k \in \mathbb{N}$ and $y = b^{(n)-}$ (by convention for any $q \geq 0$, $b^{(q)-}(0) = 0$).

We have the following preliminary fact:

Fact 4.12 Let $k \geq 1$, $l \geq n$ be two integers, and $b \in \mathcal{V}$ such that $\forall h = 1, \ldots, l$, $y^{(n)+}(k+h) = 0$
where $y = b^{(n)-}$. Then $U_b(k+l)$ is uniquely determined by $U_b(k)$, $b(k)$ and $b(k+l)$ as follows:

$$U_b(k+l) = M(l)U_b(k) + N(l)(b(k+l) - b(k))$$

where $M(l)$ is a $(2n - 2)$ square matrix and $N(l)$ a $(n - 1)$ column vector, both functions
of the integer variable $l$. The expressions of $M(l)$ and $N(l)$ as functions of $l$, are given in
appendix A.4.3.

For the sequence $(k_i)_{1 \leq i \leq p}$, let us denote $l_i = k_i - k_{i-1}$ with the convention $k_0 = 0$.

Notation 4.13 $P$ and $Q$ are the two $(n - 1) \times (2n - 2)$ matrices $P = [I_{n-1,n-1} | O_{n-1,n-1}]$ and
$Q = [O_{n-1,n-1} | I_{n-1,n-1}]$, where $I_{n-1,n-1}$ and $O_{n-1,n-1}$ are respectively the $(n - 1)^2$ identity
and null matrices.

48
Fact 4.14 Let \((R_i)_{0 \leq i \leq p}\) and \((\mathcal{M}_i)_{0 \leq i \leq p}\) be the matrix sequences recursively defined by
\[
\begin{align*}
R_0 &= P^T \\
\mathcal{M}_0 &= [ ] \quad \text{and} \quad \forall i = 1, \ldots, p, \quad R_i &= M(l_i)R_{i-1} \\
\mathcal{M}_i &= \begin{bmatrix} \mathcal{M}_{i-1} \\ \mathcal{P}R_{i-1} \end{bmatrix}
\end{align*}
\] (4.8)
Then the matrix \(PR_p\) is invertible and \((W^0_{p,:\leq i \leq p-1})\) and \(V^0_p\) can be expressed in terms of \(R_p\) and \(\mathcal{M}_p\) as follows:
\[
\begin{bmatrix} W^0_{p,0} \\ \vdots \\ W^0_{p,p-1} \end{bmatrix} = -\mathcal{M}_p(PR_p)^{-1}PN(l_p)
\] (4.9)
\[
V^0_p = \left(Q - (QR_p)(PR_p)^{-1}P\right)N(l_p)
\] (4.10)

We study in detail the case \(n = 2\) in appendix A.6

4.5.5 Recapitulation

With the resolution of \(W^0_{p,\leq}, V^0_p\), algorithm 3.3 becomes

Algorithm 3.4: In algorithm 3.3, replace lines (1) and (7) by
(1) Initialize \(p = 0, k_0 = 0, I_0 = \emptyset, a_0(0) = 0, V_0 = O_{n-1}, R_0 = P^T, \mathcal{M}_0 = [ ]\).
(7.1) Call \(l_p = k_p - k_{p-1}\).
(7.2) Take \(R_p = M(l_p)R_{p-1}\) and \(\mathcal{M}_p = \begin{bmatrix} \mathcal{M}_{p-1} \\ \mathcal{P}R_{p-1} \end{bmatrix}\).
(7.3) Calculate
\[
\begin{bmatrix} W^0_{p,0} \\ \vdots \\ W^0_{p,p-1} \end{bmatrix} = -\mathcal{M}_p(PR_p)^{-1}PN(l_p)
\]
and
\[
V^0_p = \left(Q - (QR_p)(PR_p)^{-1}P\right)N(l_p)
\]

This requires the inversion of a square matrix \((PR_p)\) of size \((n-1)^2\) at every iteration \(p\). Note that in the case \(n = 2\), this is just a scalar inversion. We also see that line (7.3) requires the memorization of the growing matrix \(\mathcal{M}_p\) of size \(p(n-1) \times (n-1)\). We saw in the previous section that \((W^0_{p,\leq i \leq p-1})\) had to be memorized as well (equivalent to a column vector of size \(p(n-1)\)).

The inversion of \(PR_p\) in relations (4.9) and (4.10) can lead to numerical difficulties as \(R_p\) is a matrix growing with \(p\) (see relation (4.8)). This problem can be solved with the following fact:

Fact 4.15 Formulas (4.9) and (4.10) are still true with the matrix sequences \((R_i)_{0 \leq i \leq p}\) and \((\mathcal{M}_i)_{0 \leq i \leq p}\) recursively defined by
\[
\begin{align*}
R_0 &= P^T \\
\mathcal{M}_0 &= [ ] \quad \forall i = 1, \ldots, p, \quad R_i &= M(l_i)R_{i-1}T_i \\
\mathcal{M}_i &= \begin{bmatrix} \mathcal{M}_{i-1} \\ \mathcal{P}R_{i-1} \end{bmatrix} T_i
\end{align*}
\] (4.11)
where \((T_i)_{0 \leq i \leq p}\) is an arbitrary sequence of invertible matrices of size \((n-1)\).
This leads to an algorithm with more computation freedom than algorithm 3.4

**Algorithm 3.5**: In algorithm 3.4, replace line (7.2) by

(7.2) - Choose an \((n - 1)\) invertible matrix \(T_p\)

\[
R_p = M(l_p)R_{p-1}T_p \quad \text{and} \quad M_p = \begin{bmatrix} M_{p-1} \\ PR_{p-1} \end{bmatrix} T_p
\]

Algorithm 3.4 will simply be the particular case of 3.5 where \(T_p\) is systematically taken equal to the identity matrix.

We propose a particular choice of \(T_p\) to make the inversion of \(PR_p\) practical. At step \(p\), let \(\mathcal{U}_p \mathcal{S}_p \mathcal{V}_p^T\) be the singular value decomposition of \(M(l_p)R_{p-1}\), where

- \(\mathcal{U}_p\) is an \((2n - 2) \times (n - 1)\) matrix with orthonormal column vectors,
- \(\mathcal{S}_p\) is an \((n - 1)^2\) invertible diagonal matrix
- \(\mathcal{V}_p\) is an \((n - 1)^2\) orthogonal matrix.

If we take \(T_p = \mathcal{V}_p \mathcal{S}_p^{-1}\), we will have

\[
R_p = (\mathcal{U}_p \mathcal{S}_p \mathcal{V}_p^T)(\mathcal{V}_p \mathcal{S}_p^{-1}) = \mathcal{U}_p.
\]

The inversion of \(PR_p = P\mathcal{U}_p\) thus becomes well conditioned. For this choice of \(T_p\), algorithm 3.5 has the particular form:

**Algorithm 3.6**: In algorithm 3.5, replace line (7.2) by

(7.2) - Perform the singular value decomposition \(\mathcal{U}_p \mathcal{S}_p \mathcal{V}_p^T\) of \(M(l_p)R_{p-1}\)

\[
R_p = \mathcal{U}_p \quad \text{and} \quad M_p = \begin{bmatrix} M_{p-1} \\ PR_{p-1} \end{bmatrix} \mathcal{V}_p \mathcal{S}_p^{-1}
\]

### 4.5.6 Algorithms 3.4, 3.5 with limited buffer

From the recursive relation of line (8) in algorithm 3.4 or 3.5, it can be derived that

\[
\forall q \geq i + 1, \quad W_{q,i} = \sum_{p=i+2}^{q} (W_{p,i} - W_{p-1,i}) + W_{i+1,i} = c_0 \sum_{p=i+1}^{q} (-1)^p \delta_p W_{p,i}^0
\]

Suppose \(p_m\) is the final number of selected indices. Then, the vectors associated with the final solution \(x_{p_m}\) are

\[
W_{p_m,i} = c_0 \sum_{p=i+1}^{p_m} (-1)^p \delta_p W_{p,i}^0 \quad \text{for} \quad 0 \leq i \leq p_m
\]

(4.12)

We show in appendix A.6.3 that, in the case \(n = 2\), \(W_{p,i}^0\) decays exponentially for a fixed \(i\) and increasing \(p\). We also observed numerically this behavior for higher orders \(n\). It can be decided that a predetermined number \(l_0\) of terms in the summation of (4.12) is enough to give a good approximation of \(W_{p_m,i}\)

\[
W_{p_m,i} \approx c_0 \sum_{p=i+1}^{i+l_0} (-1)^p \delta_p W_{p,i}^0 = W_{i+l_0,i}
\]

(4.13)

In other words, every \(W_{p,i}\) produced by line (8) should be output as soon as \(p \geq i + l_0\) or \(i \leq p - l_0\). This implies that only \([W_{p,p-l_0-1} \cdots W_{p,p-1}]\) will be needed and kept in memory.
for the next iteration step \((p + 1)\). This requires a buffer of length \((l_0 - 1)(n - 1)\). Another consequence is that, at step \(p\), \([W_{p-1}^{0} \cdots W_{p}^{0}]\) are the only values needed from line (7) of algorithm 3.4 or 3.5. This means that, in line (7.3), only the last \(l_0(n - 1)\) rows of \(M_p\) need to be known. Therefore, this requires the memorization of the last \((l_0 - 1)(n - 1)\) rows of \(M_{p-1}\) from the previous step \((p - 1)\) for the calculation of line (7.2) at step \(p\). Since the rows of \(M_{p-1}\) have length \((n - 1)\), this requires a buffer of length \((l_0 - 1)(n - 1) \times (n - 1)\). In total, we need a buffer of size \((l_0 - 1)n(n - 1)\).

Conversely, if we have a buffer with a size limited to \(s\), then the number of terms in the approximation (4.13) will be the integer part of \(\frac{s}{n(n-1)} + 1\).
Chapter 5

Numerical experiments

We performed numerical tests to both verify our analytical evaluation of the MSE and validate our projection algorithms. For convenience, but also to keep the same conditions of signal coding as in chapter 3, we dealt with periodic input signals. We actually reduced the complexity of input signals to sinusoids.

We draw $X_0$ at random in the Fourier domain and then perform an inverse DFT for different oversampling rates ($M > 2N + 1$). After calculating the code sequence $C_0$ of $X_0$ through a fixed A/D converter, we measure the MSE between $X_0$ and the estimate $X$ obtained by alternate projections. We repeat this operation on a certain number of drawn signals (300 to 1000) and calculate the statistical average of the MSE. We know that alternating projections on convex sets converges theoretically to their intersection. In practice, however, we have to limit the number of iterations to only obtain an approximation of an element in $V_0 \cap \Gamma(C_0)$.

5.1 Validation of the $R^{-(2n+2)}$ behavior of the MSE with sinusoidal inputs

Working with sinusoidal inputs, we confirm that the expectation of $MSE(X_0, X)$ decreases with $R$ proportionally to $R^{-(2n+2)}$, where $n$ is the order of the circuit. In figure 5.1 and 5.2, we show the comparison of performance in SNR between the classical reconstruction and the alternate projection method. We represent the SNR in dB, where the reference (0dB) corresponds to an MSE equal to $\frac{\sigma^2}{R^2}$ (variance of a random signal with a uniform distribution in $[-\frac{1}{2}, \frac{1}{2}]$). For classical reconstruction SNR, we used the theoretical formula in [2] giving the in-band noise power for an $n^{th}$ order converter, assuming that the quantization error signal is uncorrelated in time:

$$\frac{\pi^{2n}}{2n+1} \cdot \frac{\sigma_q^2}{R^{2n+1}}$$

(5.1)

where $\sigma_q^2$ is the variance of the quantization error power. For the alternate projection method, since we want to find an estimate as close as possible to $V_0 \cap \Gamma(C_0)$ to validate our theoretical evaluation of MSE, we stop the iteration process as soon as the relative increment of SNR becomes less than 0.01 dB.

For the case $n = 0$, we considered simple ADC with uniform quantization and input signals of peak-to-peak amplitude equal to $2q$ and random dc component, where $q$ is the step
Figure 5.1: SNR of signal reconstruction versus oversampling rate, with classical (theoretical eq.(5.1) and alternate projection method, for simple ADC and 1st to 4th order single-bit multi-stage ΣΔ modulation.

size. By taking this amplitude, we make sure that input sinusoids have at least 4 threshold crossings. As a first estimate in the iteration, we systematically took the direct conversion of the code sequence, where every sample is the center of the quantization interval indicated by the corresponding code word.

For the case \( n \geq 1 \), we considered \( n^{th} \) order single-bit multi-stage ΣΔ modulators. The output values of the feedback DACs are \( \pm \frac{q}{2} \). Input sinusoids drawn for the test have an amplitude (peak-to-peak) equal to \( \frac{q}{2} \) with a random phase and a random dc component such that the signal remains in the interval \([-\frac{q}{2}, \frac{q}{2}]\). We take as first estimate the zero signal.

Figure 5.1 shows that we systematically improve the signal reconstruction using alternate projections, regardless of the converter. Moreover, the corresponding curves of SNR (dotted lines) yield a slope of \((2n+2) \times 3 \) dB per octave of oversampling \( R \), when \( n \) is the order of the converter, while classical reconstruction yield a slope of \((2n+1) \times 3 \). The 3 dB/octave slope improvement is more obvious in figure 5.2 where we show the SNR gain over classical reconstruction. We also plot results at even higher rates \( R \) to emphasize the asymptotic increase of the gain.

5.2 Efficiency of projection algorithms for \( n \geq 2 \)

We recall that when \( n \geq 2 \), due to computation constraints, we could not deal with the direct projection on \( \Gamma'(C_0) \). In this numerical tests, we actually alternated projections between \( V_0 \) and \( \Gamma_I(C_0) \) where \( I \) does not cover the full time range. In section 4.5, we showed how the time indices of \( I \) were recursively selected according to \( C_0 \) and the estimate to project, hoping
that $\Gamma_I(C_0)$ will be close enough to $\Gamma(C_0)$. But we did not propose any analysis of "quality" of such time index range $I$, nor proved that alternating projections between $V_0$ and $\Gamma_I(C_0)$ would converge to an element of $V_0 \cap \Gamma(C_0)$. Note that $I$ is not necessarily the same at every iteration, since it depends on the estimate which is being projected. The fact is that those "non-perfect" algorithms still confirm the $M^{-(2n+2)}$ behavior of the MSE we anticipated in chapter 3. Moreover, in figures 5.1 and 5.2, the absolute improvement of SNR looks similar whether $n \geq 2$ or $n = 0, 1$ where we perform the exact projection on $\Gamma(C_0)$. At last, as shown in table 5.1, the number of iterations needed does not dramatically increase from $n = 1$ to $n = 2$.

5.3 Specific tests for 1st and 2nd order $\Sigma\Delta$ modulation

We had a closer look at 1st order $\Sigma\Delta$ modulation with constant inputs. We know from [9, 11] that when $V_0$ is reduced to constant signals (1 dimension), the MSE is lower bounded by $O(M^{-3})$. In figures 5.1 and 5.2, we have shown that a $M^{-(2n+2)}$ behavior is recovered as soon as input signals are sinusoids with some substantial amplitude ($\frac{\alpha}{2}$). Now, to better understand the transition between constants and sinusoids, we performed tests where $V_0$ is still the subspace of sinusoids, but input signals $X_0$ have amplitudes varying between 0 and $\frac{\alpha}{2}$. It is important to understand that, in this test configuration, the alternate projections will in general lead to a sinusoidal estimate $X$ even if $X_0$ is constant, since $X \in V_0 \cap \Gamma(X_0)$. However, figure 5.3 shows that the alternate projections don’t succeed in improving the $O(M^{-3})$ behavior, even though the computed estimates are no longer constrained to be constant themselves. On the other hand, as soon as $X_0$ has an amplitude larger than $\frac{\alpha}{2000}$,
Table 5.1: Number of iterations needed in alternate projections, until SNR increment per iteration is less than 0.01 dB.

<table>
<thead>
<tr>
<th>Oversampling rate: R</th>
<th>n=1</th>
<th>n=2</th>
<th>n=3</th>
</tr>
</thead>
<tbody>
<tr>
<td>16 = 4 octaves</td>
<td>17</td>
<td>21</td>
<td>38</td>
</tr>
<tr>
<td>32 = 5 octaves</td>
<td>18</td>
<td>23</td>
<td>43</td>
</tr>
<tr>
<td>64 = 6 octaves</td>
<td>21</td>
<td>26</td>
<td>36</td>
</tr>
<tr>
<td>128 = 7 octaves</td>
<td>21</td>
<td>30</td>
<td>39</td>
</tr>
</tbody>
</table>

after some “hesitations” at low oversampling rates, the $M^{-4}$ behavior is recovered.

Figure 5.4 shows the results we obtained for two kinds of 2nd order ΣΔ modulation: a double loop modulator and a two-stage modulator. When considering sinusoids of amplitude $\frac{3}{2}$, we find a $\frac{1}{N^2}$ behavior of the MSE for both types of modulation. In the case of constant inputs decoded by constant estimates, Hein and Zakhor found numerically $O(M^{-6})$ for the two-stage modulator and $O(M^{-5})$ for the double loop version. Again, since the $O(M^{-5})$ does not agree with our anticipated $\frac{1}{N^2}$ behavior in the case of double loop modulation, we looked at what the alternate projections would propose when decoding constant inputs with sinusoidal estimates. This time, as shown in figure 5.4, we keep a $O(M^{-6})$ behavior.

5.4 Further improvements with relaxation coefficients

The theorem of alternate projections by Youla [3], remains valid when each projection $P$ is replaced by the following operator:

$$P_r = P + (r - 1)(P - I)$$

where the relaxation coefficient $r$ belongs to $[0, 2[$, and $I$ is the identity operator. Note that $P_0 = I$ and $P_1 = P$. It is often experimentally observed that taking relaxation coefficients greater than 1 can accelerate the convergence process. In our notations, we call $\alpha$ and $\beta$ the coefficients we apply on the projections on $\Gamma(C_0)$ and $V_0$ respectively. We observed that, with simple ADC, we obtained the fastest convergence when $\alpha = \beta = 2$, but also the best results in terms of SNR gain. We show in figure 5.5 the improvement we obtained with regard to the regular case of projection $\alpha = \beta = 1$. With simple ADC, we even found that the iteration ends up with an estimate belonging to the interior of $V_0 \cap \Gamma(C_0)$. We also show in figure 5.5 the improvement we obtained for 1st order ΣΔ modulation with $\alpha = 2$ and $\beta = 1$.

5.5 Circuit imperfections

As shown in sections 2.2 and 2.5, we can include circuit imperfections in the description of $\Gamma(C_0)$, especially deviations in quantization and in the feedback DAC outputs. We have
Figure 5.3: SNR gain over classical reconstruction using alternate projections, with 1st order \( \Sigma \Delta \) modulation and sinusoidal inputs of different amplitudes. Even with sinusoids of amplitude \( q/2000 \), the gain of 3dB/octave is recovered for \( R \) high enough.

performed these tests with multi-stage single-bit \( \Sigma \Delta \) modulations, which are known to be particular sensitive to feedback nonlinearities. We measure the level of deviation in terms of percentage of the step size \( q \). In figure 5.6, we see that 1% of imperfections does not affect the performances with regard to the ideal case. This means that, if the maximum of excursion of input signals is 1V, 10mV of deviation will not decrease the quality of reconstruction when using alternate projections. We of course assume that these deviations can be measured on the circuit which is being considered. When imperfections are of 10% (100mV!), we still have results similar to an ideal converter when using a classical reconstruction.
Figure 5.4: SNR gain over classical reconstruction for 2nd order $\Sigma \Delta$ modulation with constant and sinusoidal inputs.

Figure 5.5: Improvement of signal reconstruction using relaxation coefficients in simple ADC and 1st order $\Sigma \Delta$ modulation.
Figure 5.6: Signal reconstruction with imperfections in quantization and DAC feedback on $2^{nd}$ and $3^{rd}$ order single-bit multi-stage $\Sigma \Delta$ modulation. With 1% of imperfections, the alternate projections are not affected. With 10%, performances are still as good as classical reconstruction with an ideal converter.
Appendix A

Appendix

A.1 Proof lemmas 3.2 and 3.5

For simplicity, we suppose that the period $T_0$ of input signals is equal to 1 (the time dimension can always be rescaled). To prove lemmas 3.2 and 3.5, the idea is to see that $\frac{V(k)}{M^k}$ can be uniformly approximated by the $n^{th}$ order integral $U^{(-n)}[t]$ of $U[p]$ taken at time $t = \frac{k}{M}$. This approximation will be expressed in lemmas A.2 ($n = 1$) and lemma A.5 ($n \geq 1$). Then the maximum value and the average of $|V(k)|$ will be approximated by the infinite and the $L_1$ norms, respectively, of this integral. The upper and lower bounds will be then derived from norm equivalence (lemma A.6).

A.1.1 Preliminary notations and lemmas

**Notation A.1** Norms $\| \cdot \|_\infty$ and $\| \cdot \|_1$

For a continuous function $f$ on $[0, 1]$, $\|f\|_\infty = \max_{t \in [0, 1]} |f[t]|$ and $\|f\|_1 = \int_0^1 |f[t]| dt$.

**Lemma A.2** Let $f$ be a function $C^1$ on $[0, 1]$. Then $\forall M \geq 1, \forall k = 1, ..., M$,

\[
\left| \frac{1}{M} \sum_{j=1}^{k} f\left[ \frac{j}{M} \right] - \int_0^1 f[t] dt \right| \leq \frac{1}{M} \|f'\|_\infty \quad (A.1)
\]

\[
\left| \frac{1}{M} \sum_{j=1}^{k} \left| f\left[ \frac{j}{M} \right] \right| - \int_0^1 |f[t]| dt \right| \leq \frac{1}{M} \|f'\|_\infty \quad (A.2)
\]

This is proved in A.1.4. We are going to generalize this lemma to higher order integration. We first introduce some notation for the $n^{th}$ order continuous time and discrete integration.

**Notation A.3** $n^{th}$ order derivative and integral

For a function $F$, $C^\infty$ on $[0, 1]$, and an integer $n \geq 0$, $f^{(n)}$ designates the $n^{th}$ order derivative and $f^{(-n)}$ is the $n^{th}$ order integral which is equal to 0 at $t = 0$. More precisely, $f^{(0)} = f$ and for $n \geq 1$, $f^{(-n)}$ is the integral of $f^{(-n+1)}$ which is equal to 0 at $t = 0$. 

59
Notation A.4 \( n \)th order discrete sum
For a function \( f \) defined on \([0,1]\), and an integer \( M \geq 1 \), \( I^n_{f,M} \) is the \( M \)-point sequence defined recursively as follows

\[
\forall k = 1, \ldots, M, \quad I^n_{f,M}(k) = f\left[k\right] \quad \text{and for} \quad n \geq 1, \quad \forall k = 1, \ldots, M, \quad I^n_{f,M}(k) = \sum_{j=1}^{k} I^{n-1}_{f,M}(j)
\]

Lemma A.5 Let \( f \) be a function \( C^1 \) on \([0,1]\) and \( n \geq 0 \) an integer.

\[
\forall M \geq 1, \forall k = 1, \ldots, M, \quad \left| \frac{1}{M^n} \sum_{j=1}^{k} I^n_{f,M}(j) - \int_{0}^{1} f(-n)[t]dt \right| \leq \frac{1}{M} \sum_{m=-n+1}^{1} \left\| f^{(m)} \right\|_{\infty} \tag{A.3}
\]

\[
\forall M \geq 1, \forall k = 1, \ldots, M, \quad \left| \frac{1}{M^n} \sum_{j=1}^{k} I^n_{f,M}(j) - \int_{0}^{1} f(-n)[t]dt \right| \leq \frac{1}{M} \sum_{m=-n+1}^{1} \left\| f^{(m)} \right\|_{\infty} \tag{A.4}
\]

This is proved in A.1.5.

Remark: Replacing \( n \) by \( n - 1 \), (A.3) can be rewritten

\[
\forall M \geq 1, \forall k = 1, \ldots, M, \quad \left| \frac{1}{M^n} I^n_{f,M}(k) - f(-n)\left[k\right] \right| \leq \frac{1}{M} \sum_{m=-n+2}^{1} \left\| f^{(m)} \right\|_{\infty} \tag{A.5}
\]

We recall that \( S \) is the unit sphere of \( V_0 \): \( S = \left\{ U \in V_0 \mid \left\| U \right\| = 1 \right\} \)

Lemma A.6 Let \( N \) be a norm on \( V_0 \) and \( n \in Z \). Then

\[
\exists B_+ > 0, \quad \forall U \in S, \quad N\left(U^{(n)}\right) \leq B_+ \tag{A.6}
\]

If moreover \( n \leq 0 \), then

\[
\exists B_- > 0, \quad \forall U \in S, \quad N\left(U^{(n)}\right) \geq B_- \tag{A.7}
\]

Proof: Properties of finite dimensional normed spaces \( \Box \)

A.1.2 Proof of lemma 3.2

By definition of \( L \), we have exactly \( V(k) = I^n_{U,M}(k) \) for \( k = 1, \ldots, M \). If we apply (A.5) on \( f = U \), we find

\[
\forall M \geq 1, \forall k = 1, \ldots, M, \quad \left| \frac{V(k)}{M^n} - U^{(-n)}\left[k\right] \right| \leq \frac{1}{M} \sum_{m=-n+2}^{1} \left\| U^{(m)} \right\|_{\infty} \tag{A.8}
\]

Let \( t_0 \in [0,1] \) be the instant when \( \left| U^{(-n)}[t] \right| \) achieves its maximum. We have

\[
\left| U^{(-n)}[t_0] \right| = \left\| U^{(-n)} \right\|_{\infty}
\]

60
For \( k \in \{1, ..., M\} \), using Lagrange inequality between \( \frac{k}{M} \) and \( t_0 \), we find
\[
\left| U^{(-n)} \left[ \frac{k}{M} \right] - U^{(-n)} \left[ t_0 \right] \right| \leq \left| \frac{k}{M} - t_0 \right| \left\| U^{(-n+1)} \right\|_{\infty}
\]
Together with (A.8), we find that \( \forall M \geq 1, \forall k = 1, ..., M \),
\[
\left| \frac{V(k)}{M^n} - U^{(-n)} \right| \leq \left| \frac{k}{M} - t_0 \right| \left\| U^{(-n+1)} \right\|_{\infty} + \frac{1}{M} \sum_{m=-n+2}^{1} \left\| U^{(m)} \right\|_{\infty} \tag{A.9}
\]
Since \( \left| \frac{k}{M} - t_0 \right| \leq 1 \) and \( \frac{1}{M} \leq 1 \), this immediately gives an upper bound to \( \left| \frac{V(k)}{M^n} \right| : \)
\[
\left| \frac{V(k)}{M^n} \right| \leq \left\| U^{(-n)} \right\|_{\infty} + \left\| U^{(-n+1)} \right\|_{\infty} + \frac{1}{M} \sum_{m=-n+2}^{1} \left\| U^{(m)} \right\|_{\infty} = \frac{1}{M} \left\| U^{(m)} \right\|_{\infty}
\]
This last term can be bounded by a constant \( c_2 > 0 \), according to relation (A.6) of lemma A.6. This proves that
\[
\max_{1 \leq k \leq M} \left| V(k) \right| \leq c_2 M^n
\]
Inequality (A.9) also implies that
\[
\left| \frac{V(k)}{M^n} \right| \geq \left\| U^{(-n)} \right\|_{\infty} - \left| \frac{k}{M} - t_0 \right| \left\| U^{(-n+1)} \right\|_{\infty} - \frac{1}{M} \sum_{m=-n+2}^{1} \left\| U^{(m)} \right\|_{\infty}
\]
For every \( M \geq 1 \), it is always possible to find \( k_M \in \{1, ..., M\} \) such that \( \left| \frac{k}{M} - t_0 \right| \leq \frac{1}{M} \). Then
\[
\max_{1 \leq k \leq M} \left| \frac{V(k)}{M^n} \right| \geq \left| \frac{V(k_M)}{M^n} \right| \geq \left\| U^{(-n)} \right\|_{\infty} - \frac{1}{M} \sum_{m=-n+1}^{1} \left\| U^{(m)} \right\|_{\infty}
\]
From relation (A.7) of lemma A.6, we can find \( B_1 > 0 \) such that \( \left\| U^{(-n)} \right\|_{\infty} \geq B_1 \). From relation (A.6), we can find \( B_2 > 0 \) such that
\[
\sum_{m=-n+1}^{1} \left\| U^{(m)} \right\|_{\infty} \leq B_2 \tag{A.10}
\]
As soon as \( M \geq \frac{2B_2}{B_1} \), then
\[
\max_{1 \leq k \leq M} \left| \frac{V(k)}{M^n} \right| \geq B_1 - \frac{B_2}{M} \geq \frac{B_1}{2}
\]
Taking \( c_1 = \frac{B_1}{2} \), then we have shown that when \( M \) is large enough
\[
\max_{1 \leq k \leq M} \left| V(k) \right| \geq c_1 M^n \quad \Box
\]
A.1.3 Proof of lemma 3.5

Applying (A.4) with \( f = U \) and \( k = M \), we find

\[
\left| \frac{1}{M} \sum_{j=1}^{M} \frac{V(k)}{M^n} - \left\| U^{(n)} \right\|_1 \right| \leq \frac{1}{M} \sum_{m=-n+1}^{1} \left\| U^{(m)} \right\|_{\infty} \tag{A.11}
\]

According to (A.10), we can bound the right hand side by \( \frac{B_2}{M} \). Consequently,

\[
\left\| U^{(-n)} \right\|_1 - \frac{B_2}{M} \leq \frac{1}{M} \sum_{j=1}^{M} \left| \frac{V(k)}{M^n} \right| \leq \left\| U^{(-n)} \right\|_1 + \frac{B_2}{M} \tag{A.12}
\]

Applying property (A.7) of lemma A.6, it is possible to find constants \( 0 < B_- < B_+ \) such that

\[
B_- \leq \left\| U^{(-n)} \right\|_1 \leq B_+.
\]

The right hand side of (A.12) can be bounded by \( c_4 = B_+ + B_2 \). Then for any \( M \geq \frac{B_+}{B_-} \), the left hand side of (A.12) can be lower bounded by

\[
c_3 = B_- - \left( \frac{B_2}{B_-} \right)^{-1} B_2 = \frac{B_-}{2} \geq 0 \quad \square
\]

A.1.4 Proof of lemma A.2

Let us consider integers \( M \geq 1 \) and \( k \in \{1, \ldots, M\} \).

Since \( f \) is continuous

\[
\forall j = 1, \ldots, k, \exists t_j \in \left[ \frac{j-1}{M}, \frac{j}{M} \right], \quad \int_{\frac{j-1}{M}}^{\frac{j}{M}} f(t) dt = \frac{1}{M} f[t_j]
\]

Then

\[
\frac{1}{M} \sum_{j=1}^{k} f\left[ \frac{j}{M} \right] - \int_{0}^{\frac{k}{M}} f(t) dt = \sum_{j=1}^{k} \left( \frac{1}{M} f\left[ \frac{j}{M} \right] - \int_{\frac{j-1}{M}}^{\frac{j}{M}} f(t) dt \right) = \frac{1}{M} \sum_{j=1}^{k} \left( f\left[ \frac{j}{M} \right] - f[t_j]\right) \tag{A.13}
\]

Since \( |f| \) is also continuous, we can derive in a similar way that

\[
\forall j = 1, \ldots, k, \exists t_j' \in \left[ \frac{j-1}{M}, \frac{j}{M} \right], \quad \frac{1}{M} \sum_{j=1}^{k} |f\left[ \frac{j}{M} \right]| - \int_{0}^{\frac{k}{M}} |f(t)| dt = \frac{1}{M} \sum_{j=1}^{k} \left( |f\left[ \frac{j}{M} \right]| - |f[t_j']| \right) \tag{A.14}
\]

For a fixed \( j \in \{1, \ldots, k\} \), \( \forall t \in \left[ \frac{j-1}{M}, \frac{j}{M} \right] \)

\[
\left| f\left[ \frac{j}{M} \right] - f(t) \right| \leq \left| f\left[ \frac{j}{M} \right] - f[t_j'] \right| \leq \frac{1}{M} \left( \max_{s \in \left[ \frac{j-1}{M}, \frac{j}{M} \right]} |f'(s)| \right) \leq \frac{1}{M} \left\| f' \right\|_{\infty}
\]

Therefore, since \( k \leq M \), both expressions (A.13) and (A.14) can be bounded in absolute value by \( \frac{1}{M} \left\| f' \right\|_{\infty} \quad \square \)
A.1.5 Proof of lemma A.5

For a fixed integer \( M \geq 1 \), let us introduce the proposition

\[
P(n) : \quad \forall k = 1, ..., M, \quad \left| \frac{1}{M^{n+1}} \sum_{j=1}^{k} I_{f,M}^{n}(j) - \int_{0}^{\frac{k}{M^{n}}} f^{(-n)}[t]dt \right| \leq \frac{1}{M} \sum_{m=-n+1}^{1} \|f^{(m)}\|_{\infty}
\]

Let us show \( P(n) \) by induction. \( P(0) \) is just the expression of lemma A.2. Suppose now that \( P(n-1) \) is true for a certain \( n \geq 1 \). Using the fact that

\[
I_{f,M}^{n}(j) = \sum_{i=1}^{j} I_{f,M}^{n-1}(i) \quad \text{and} \quad f^{(-n)}[\frac{j}{M^{n}}] = \int_{0}^{\frac{j}{M^{n}}} f^{(-n+1)}[t]dt
\]

we can write

\[
\frac{1}{M^{n+1}} \sum_{j=1}^{k} I_{f,M}^{n}(j) - \int_{0}^{\frac{k}{M^{n}}} f^{(-n)}[t]dt = \frac{1}{M} \sum_{j=1}^{k} \left( \frac{1}{M^{n}} \sum_{i=1}^{j} I_{f,M}^{n-1}(i) - \int_{0}^{\frac{j}{M^{n}}} f^{(-n+1)}[t]dt \right)
\]

\[
+ \left( \frac{1}{M} \sum_{j=1}^{k} f^{(-n)}[\frac{j}{M^{n}}] - \int_{0}^{\frac{j}{M^{n}}} f^{(-n)}[t]dt \right)
\]

Applying the fact that \( P(n-1) \) is true, we find

\[
\frac{1}{M} \sum_{j=1}^{k} \frac{1}{M^{n}} \sum_{i=1}^{j} I_{f,M}^{n-1}(i) - \int_{0}^{\frac{j}{M^{n}}} f^{(-n+1)}[t]dt \leq \frac{k}{M^{2}} \sum_{m=-n+2}^{1} \|f^{(m)}\|_{\infty} \leq \frac{1}{M} \sum_{m=-n+2}^{1} \|f^{(m)}\|_{\infty}
\]

Then applying (A.1) on function \( f^{(-n)} \) which is \( C^1 \), we find

\[
\left| \frac{1}{M} \sum_{j=1}^{k} f^{(-n)}[\frac{j}{M^{n}}] - \int_{0}^{\frac{j}{M^{n}}} f^{(-n)}[t]dt \right| \leq \frac{1}{M} \|f^{(-n+1)}\|_{\infty}
\]

\( P(n) \) is then a consequence of (A.16),(A.17) and (A.18). The induction is completed: \( P(n) \) is true for every \( n \geq 0 \).

To show (A.4), we write an equation similar to (A.16), using (A.15)

\[
\frac{1}{M^{n+1}} \sum_{j=1}^{k} I_{f,M}^{n}(j) - \int_{0}^{\frac{k}{M^{n}}} f^{(-n)}[t]dt = \frac{1}{M} \sum_{j=1}^{k} \left( \frac{1}{M^{n}} \sum_{i=1}^{j} I_{f,M}^{n-1}(i) - \int_{0}^{\frac{j}{M^{n}}} f^{(-n+1)}[t]dt \right)
\]

\[
+ \left( \frac{1}{M} \sum_{j=1}^{k} f^{(-n)}[\frac{j}{M^{n}}] - \int_{0}^{\frac{j}{M^{n}}} f^{(-n)}[t]dt \right)
\]

Using the inequality \(|a| - |b| \leq |a - b|\) and applying \( P(n-1) \), the first term can be bounded in absolute value like in (A.17). The second term can be bounded by \( \frac{1}{M} \|f^{(-n+1)}\|_{\infty} \) by applying (A.2) on \( f^{(-n)} \) \( \square \)

63
A.2 Proof of fact 4.5

Let $D_{\pm}^{(n)}$ and $D_{\pm}^{(n)}$ be the linear and time invariant operators whose $z$ transforms are $(1-z^{-1})^n$ and $z^{-1})^n$. For two given sequences $a$ and $x$, let $\tilde{a}$ and $\tilde{x}$ be their respective extension to negative time indices:

$$\forall k \geq 1, \quad \tilde{a}(k) = a(k) \quad \text{and} \quad \tilde{x}(k) = x(k)$$

$$\forall k \leq 0, \quad \tilde{a}(k) = \tilde{x}(k) = 0$$

It can be verified that $a^{(n)}-\tilde{a}$ and $x^{(n)}-\tilde{x}$ are the restriction to $\mathbb{N}^*$ of $D_{\pm}^{(n)}[\tilde{a}]$ and $D_{\pm}^{(n)}[\tilde{x}]$. Let $d_{-}^{(n)}$ and $d_{+}^{(n)}$ be the impulse responses of $D_{\pm}^{(n)}$. We have

$$\forall j \geq 1, \quad a^{(n)}-(j) = \left( d_{-}^{(n)} * \tilde{a} \right)(j) = \sum_{k=1}^{+\infty} d_{-}^{(n)}(k-j)a(k) \quad (A.19)$$

$$\forall k \geq 1, \quad x^{(n)}+(k) = \left( d_{+}^{(n)} * \tilde{x} \right)(k) = \sum_{j=1}^{+\infty} d_{+}^{(n)}(j-k)x(j) \quad (A.20)$$

Suppose that $x = L^{-1}[a] = a^{(n)}$. Then $\phi(a) = \frac{1}{2} \sum_{j \geq 1} |x(j)|^2$ and

$$\frac{\partial \phi}{\partial a}(k) = \frac{1}{2} \sum_{j=1}^{+\infty} x(j) \frac{\partial |x(j)|^2}{\partial a(k)} = \sum_{j=1}^{+\infty} \frac{\partial x(j)}{\partial a(k)} x(j) \quad (A.21)$$

Using (A.19), we have

$$\frac{\partial x(j)}{\partial a(k)} = \frac{\partial a^{(n)}(j)}{\partial a(k)} = d_{-}^{(n)}(k-j) \quad (A.22)$$

Since the $z$ transforms of $D_{\pm}^{(n)}$ and $D_{\pm}^{(n)}$ verify $D_{\pm}^{(n)}(z) = (-1)^n D_{\pm}^{(n)}(z^{-1})$, then

$$\forall k \in \mathbb{Z}, \quad d_{-}^{(n)}(k) = (-1)^n d_{+}^{(n)}(-k) \quad (A.23)$$

Using successively (A.21), (A.22), (A.23) and (A.20) we find

$$\frac{\partial \phi}{\partial a}(k) = \sum_{j=1}^{+\infty} d_{-}^{(n)}(k-j)x(j) = (-1)^n \sum_{j=1}^{+\infty} d_{+}^{(n)}(j-k)x(j) = (-1)^n x^{(n)}+(k) \quad \square$$

A.3 Proof of fact 4.7

Using Euler characterization and unicity of the minimum, we are going to show that

$$s_0 = a_{p-1} + \epsilon \delta b_p$$

is necessarily the minimum of $\phi$ subject to $\mathcal{C}(I_p)$.

First, we have to check that $s_0 \in \mathcal{C}(I_p)$. We use the fact that

$$\forall i = 1, \ldots, p-1, \quad b_p(k_i) = 0 \quad \text{and} \quad b_p(k_p) = 1$$
This implies that \( s_0(k_i) = a_{p-1}(k_i) \in Q(k_i) \), since \( a_{p-1} \in C(I_{p-1}) \). We also find that \( s_0(k_p) = a_{p-1}(k_p) + \epsilon \delta \). Using assumption (2) on \( a_{p-1} \) and the definition of \( \delta \) in (4), we conclude that
\[
s_0(k_p) \text{ is the closest bound of } Q(k_p) \text{ to } a_{p-1}(k_p)
\] (A.24)

Therefore \( s_0(k_p) \in Q(k_p) \).

Let us now show that the Euler inequality characterization is satisfied by \( s_0 \) with regard to \( C(I_p) \).

From fact 4.5, we can see that \( a \mapsto \frac{\partial \phi}{\partial a} \) is linear. Therefore
\[
\frac{\partial \phi}{\partial s_0} = \frac{\partial \phi}{\partial a_{p-1}} + \epsilon \delta \frac{\partial \phi}{\partial b_p} \tag{A.25}
\]

Using the fact that \( C(I_{p-1}) \) and \( E(I_p) \) are separable constraints and using theorem 4.3, we find that
\[
\forall k \notin I_{p-1}, \quad \frac{\partial \phi}{\partial a_{p-1}}(k) = 0 \tag{A.26}
\]
\[
\forall k \notin I_p, \quad \frac{\partial \phi}{\partial b_p}(k) = 0 \tag{A.27}
\]

Since \( I_p \supset I_{p-1} \), this implies that
\[
\forall k \notin I_p, \quad \frac{\partial \phi}{\partial s_0}(k) = 0 \tag{A.28}
\]

Let us now study the sign of \( \frac{\partial \phi}{\partial s_0}(k) \cdot (a(k) - s_0(k)) \) when \( k \in I_p \) and \( a \in C(I_p) \).

We first consider the case \( k = k_p \). Applying (A.26) and (4.4) at \( k = k_p \), we find \( \frac{\partial \phi}{\partial a_{p-1}}(k_p) = 0 \) and \( \frac{\partial \phi}{\partial b_p}(k_p) \geq 0 \). From (A.25), this implies that \( \frac{\partial \phi}{\partial s_0}(k_p) = \epsilon \delta \frac{\partial \phi}{\partial b_p}(k_p) \) and therefore
\[
\text{sign} \left( \frac{\partial \phi}{\partial s_0}(k_p) \right) = \epsilon \tag{A.29}
\]

For any \( r \in Q(k_p) \), we have the same sign equal to \( \epsilon \), according to assumption (2) and (A.24). We can apply this to \( r = a_{p-1}(k_p) \in Q(k_p) \) and find that
\[
\frac{\partial \phi}{\partial s_0}(k_p) \cdot (a(k_p) - s_0(k_p)) \geq 0 \tag{A.30}
\]

Next, we consider \( k = k_i \), where \( 1 \leq i \leq p-1 \). Because \( a \in C(I_p) \) also belongs to \( C(I_{p-1}) \), we can apply theorem 4.3 on \( S = C(I_{p-1}) \) and find that
\[
\frac{\partial \phi}{\partial a_{p-1}}(k_i) \cdot (a(k_i) - a_{p-1}(k_i)) \geq 0 \tag{A.31}
\]

Using assumption (2) and (4.4), the signs of \( \frac{\partial \phi}{\partial a_{p-1}}(k_i) \) and \( \frac{\partial \phi}{\partial b_p}(k_i) \) are respectively \( \epsilon(-1)^{p-i} \) and \( (-1)^{p-i} \). From (A.25), we derive that
\[
\text{sign} \left( \frac{\partial \phi}{\partial s_0}(k_i) \right) = \epsilon(-1)^{p-i} \tag{A.32}
\]
Since \( a_p(k_i) = s_0(k_i) \), (A.31) implies
\[
\frac{\partial \phi}{\partial s_0}(k_i) \cdot (a(k_i) - s_0(k_i)) \geq 0 \quad (A.33)
\]
Gathering (A.28), (A.30) and (A.33), we have
\[
\sum_k \frac{\partial \phi}{\partial s_0}(k) \cdot (a(k) - s_0(k)) \geq 0
\]
Property \((\beta)\) is a consequence of (A.29) and (A.32) \(\square\)

A.4 Proof of fact 4.12

Let \( k \geq 1 \), \( l \geq n \) be two integers, and \( b \in V \) such that \( \forall h = 1, \ldots, l, \ y^{(n)+}(k + h) = 0 \) where \( y = b^{(n)} \).

A.4.1 Preliminaries

**Definition A.7** We define \( S_p(k) \) as follows
\[
S_0(k) = \begin{cases} 
1 & \text{for } k \geq 1 \\
0 & \text{for } k \leq 0 \end{cases} \quad \forall p \geq 0 , \ S_{p+1}(k) = \begin{cases} 
\sum_{l=1}^{k} S_p(l) & \text{for } k \geq 1 \\
0 & \text{for } k \leq 0 
\end{cases} \quad (A.34)
\]

We have an analytical expression of \( S_p(k) \), as follows

**Fact A.8**
\[
\forall p \geq 1, k \geq 1 , \ S_p(k) = \frac{k(k+1)...(k+p-1)}{p!} \quad (A.35)
\]

Proof: This is obtained by induction with \( p \), using definition A.7 \(\square\)

**Fact A.9** For \( p \geq 0 \), and \( h_1, h_2 \) two integers such that \( 0 \leq h_1 \leq h_2 \), then
\[
\sum_{r=1}^{h_2} S_p(r - h_1) = S_{p+1}(h_2 - h_1)
\]

Proof: With the change of variable \( r' = r - h_1 \), using the fact that \( 1 - h_1 \leq 1 \) and definition A.7, we find
\[
\sum_{r=1}^{h_2} S_p(r - h_1) = \sum_{r' = 1 - h_1}^{h_2 - h_1} S_p(r') = \sum_{r' = 1}^{h_2 - h_1} S_p(r') = S_{p+1}(h_2 - h_1) \quad \square
\]

**Notation A.10** \( S_p^{-1}(k) = (S_p(k))^{-1} \)

66
Fact A.11  \( \forall q \in \{1, ..., n\}, \ \forall h = 1, ..., l + q - 1, \)
\[
y^{(n-q)+(k+h)} = \sum_{j=0}^{q-1} S_j(h-j)y^{(n-q+j)+(k+1)} \tag{A.36}
\]
The proof is shown in section A.4.4.

Fact A.12  \( \forall q \in \{1, ..., n\}, \ \forall h = 1, ..., l, \)
\[
a^{(n-q)-(k+h)} = \sum_{j=0}^{q-1} S_j(h)b^{(n-q+j)-(k)} + \sum_{j=0}^{n-1} S_{q+j}(h-j)y^{(j)+(k+1)} \tag{A.37}
\]
The proof is shown in section A.4.5.

A.4.2 Proof of fact 4.12

Considering \( q = 2, ..., n \), relation (A.36) of fact A.11 can be written in matrix way:
\[
\forall h = 1, ..., l
\]
\[
\begin{bmatrix}
y^{(n-2)+(k+h)} \\
y^{(n-3)+(k+h)} \\
\vdots \\
y^{(0)+(k+h)}
\end{bmatrix}
= M_1(h) \cdot
\begin{bmatrix}
y^{(n-2)+(k+1)} \\
y^{(n-3)+(k+1)} \\
\vdots \\
y^{(0)+(k+1)}
\end{bmatrix}
+ N_1(h) \cdot y^{(n-1)+(k+1)} \tag{A.38}
\]
where \( M_1(h) \) is the \((n-1)\) square matrix given in (A.42) and \( N_1(h) \) is the \((n-1)\) column vector given in (A.46). Similarly, considering \( q = 1, ..., n - 1 \), relation (A.37) of fact A.12 can be written in matrix way:
\[
\forall h = 1, ..., l
\]
\[
\begin{bmatrix}
a^{(n-1)-(k+h)} \\
a^{(n-2)-(k+h)} \\
\vdots \\
a^{(1)-(k+h)}
\end{bmatrix}
= M_2(h) \cdot
\begin{bmatrix}
a^{(n-1)-(k)} \\
a^{(n-2)+(k)} \\
\vdots \\
a^{(1)-(k)}
\end{bmatrix}
+ M_3(h) \cdot
\begin{bmatrix}
y^{(n-2)+(k+1)} \\
y^{(n-3)+(k+1)} \\
\vdots \\
y^{(0)+(k+1)}
\end{bmatrix}
+ N_2(h) \cdot y^{(n-1)+(k+1)} \tag{A.39}
\]
where \( M_2(h) \) and \( M_3(h) \) are the \((n-1)\) square matrices given in (A.43) and (A.44) respectively and \( N_2(h) \) is the \((n-1)\) column vector given in (A.47). Using the notation \( U_b(k) \) (notation 4.13), the relations (A.38) and (A.39) can be summarized as:
\[
\forall h = 1, ..., l, \quad U_b(k+h) = M_0(h)U_b(k) + \begin{bmatrix}
N_1(h) \\
N_2(h)
\end{bmatrix} y^{(n-1)+(k+1)} \tag{A.40}
\]
where \( M_0(h) \) is the \((2n-2)\) square matrix given in (A.45). Relation (A.37) applied at \( q = n \) can be written:
\[
\forall h = 1, ..., l,
\]
\[
a(k+h) = b(k) + \left[N_3(h) N_4(h)\right]U_b(k) + S_{2n-1}(h - (n - 1))y^{(n-1)+(k+1)} \tag{A.41}
\]
67
where $N_3(h)$ and $N_4(h)$ are the $(n - 1)$ row vectors given in (A.48) and (A.49) respectively. When taking $h = l$, $S_{2n-1}(l - (n - 1)) \neq 0$ since $l \geq n$. We can therefore eliminate $y^{(n-1)+1}(k + 1)$ between (A.40) and (A.41) at $h = l$. We find

$$U_b(k + l) = M(l)U_b(k) + N(l)(b(k + l) - b(k))$$

where $M(l)$ is the $(2n - 2)$ square matrix given in (A.50) and $N(l)$ is the $(2n - 2)$ column vector given in (A.51)

**A.4.3 Analytical expressions of $M(l)$ and $N(l)$ versus $l$**

We recapitulate the construction of matrices $M(l)$ and $N(l)$. From fact A.8, we have

$$\forall p \geq 1, \quad S_p(k) = 0 \text{ for } k \leq 0 \quad \text{and} \quad S_p(k) = \frac{k(k+1)\ldots(k+p-1)}{p!} \text{ for } k \geq 1$$

In paragraph A.4.2, we have constructed

$$M_1(l) = \begin{bmatrix}
S_0(l+1) & 0 & \cdots & 0 \\
S_1(l) & S_0(l+1) & 0 & \vdots \\
\vdots & \ddots & \ddots & \ddots \\
S_{n-2}(l - (n - 3)) & \cdots & S_1(l) & S_0(l+1)
\end{bmatrix} \quad (A.42)$$

$$M_2(l) = \begin{bmatrix}
S_0(l) & 0 & \cdots & 0 \\
S_1(l) & S_0(l) & 0 & \vdots \\
\vdots & \ddots & \ddots & \ddots \\
S_{n-2}(l) & \cdots & S_1(l) & S_0(l)
\end{bmatrix} \quad (A.43)$$

$$M_3(l) = \begin{bmatrix}
S_{n-1}(l - (n - 2)) & \cdots & S_2(l - 1) & S_1(l) \\
S_{n-2}(l - (n - 2)) & \cdots & S_3(l - 1) & S_2(l) \\
\vdots & \ddots & \ddots & \ddots \\
S_{2n-3}(l - (n - 2)) & \cdots & S_n(l - 1) & S_{n-1}(l)
\end{bmatrix} \quad (A.44)$$

$$M_0(l) = \begin{bmatrix}
M_1(l) & 0 \\
M_3(l) & M_2(l)
\end{bmatrix} \quad (A.45)$$

$$N_1(l) = \begin{bmatrix}
S_1(l) & S_2(l - 1) & \cdots & S_{n-1}(l - (n - 2))
\end{bmatrix}^T \quad (A.46)$$

$$N_2(l) = \begin{bmatrix}
S_n(l - (n - 1)) & \cdots & S_{2n-2}(l - (n - 1))
\end{bmatrix}^T \quad (A.47)$$

$$N_3(l) = \begin{bmatrix}
S_{2n-2}(l - (n - 2)) & \cdots & S_{n-1}(l - 1) & S_n(l)
\end{bmatrix} \quad (A.48)$$

$$N_4(l) = \begin{bmatrix}
S_{n-1}(l) & \cdots & S_1(l)
\end{bmatrix} \quad (A.49)$$

$$M(l) = M_0(l) - S_{2n-1}^{-1}(l - (n - 1)) \begin{bmatrix}
N_1(l) \\
N_2(l)
\end{bmatrix} \begin{bmatrix}
N_3(l) \\
N_4(l)
\end{bmatrix} \quad (A.50)$$

68
\[ N(l) = S_{2n-1}^{-1} (l - (n - 1)) \begin{bmatrix} N_1(l) \\ N_2(l) \end{bmatrix} \]  
(A.51)

Once \( n \) is fixed, \( M(l) \) and \( N(l) \) can be precalculated in terms of \( l \). We give their expression for \( n = 2 \) in appendix A.6.1

**A.4.4 Proof of fact A.11**

We will use the fact that

\[ \forall j \geq 0, \forall h \geq 1, \quad y^{(j)+}(k + h) = y^{(j)+}(k + 1) + \sum_{r=1}^{h-1} y^{(j+1)+}(k + r) \]  
(A.52)

which is a consequence of the relation \( y^{(\ell+1)+}(h) = y^{(\ell)+}(h + 1) - y^{(\ell)+}(h) \). We are going to prove (A.36) by induction with \( q \in \{0, \ldots, n\} \).

By assumption, \( \forall h = 1, \ldots, l - 1, \quad y^{(n)+}(k + h) = 0. \) Applying (A.52) with \( j = n - 1, \) we find that \( \forall h = 1, \ldots, l, \)

\[ y^{(n-1)+}(k + h) = y^{(n-1)+}(k + 1) = S_0(h) y^{(n-1)+}(k + 1) \]

since \( S_0(h) = 1. \) This proves (A.36) for \( q = 1. \)

Now let us assume that (A.36) is true for a certain \( q \in \{1, \ldots, n - 1\} \). Let us consider \( h \in \{1, \ldots, l + q\} \). Applying (A.52) with \( j = n - q - 1 \) we find

\[ y^{(n-q-1)+}(k + h) = y^{(n-q-1)+}(k + 1) + \sum_{r=1}^{h-1} y^{(n-q)+}(k + r) \]  
(A.53)

Using the assumption that (A.36) is true at \( q, \) we find that

\[ \sum_{r=1}^{h-1} y^{(n-q)+}(k + r) = \sum_{r=1}^{h-1} \left( \sum_{j=0}^{q-1} S_j(r - j) y^{(n-q+j)+}(k + 1) \right) \]

\[ = \sum_{j=0}^{q-1} \left( \sum_{r=1}^{h-1} S_j(r - j) \right) y^{(n-q+j)+}(k + 1) \]

Using fact A.9 and the fact that \( S_0(h) = 1, \) (A.53) becomes

\[ y^{(n-q-1)+}(k + h) = S_0(h) y^{(n-q-1)+}(k + 1) + \sum_{j=0}^{q-1} S_{j+1}(h - 1 - j) y^{(n-q+j)+}(k + 1) \]

With the change of variable \( j' = j + 1 \) and inserting the first term into the sum, we find that, for \( h = 1, \ldots, l + (q + 1) - 1 \)

\[ y^{(n-(q+1))}(k + h) = \sum_{j'=0}^{(q+1)-1} S_{j'}(h - j') x^{(n-(q+1)+j')}(k + 1) \]

which proves (A.36) for \( (q + 1) \). The induction is completed. \( \Box \)
A.4.5 Proof of fact A.12

We will use the fact that
\[ \forall j \geq 0, \ h \geq 1, \ b^{(j)}-(k + h) = b^{(j)}-(k) + \sum_{r=1}^{h} b^{(j+1)}-(k + r) \tag{A.54} \]

which is consequence of the relation \( b^{(j+1)}+(h) = b^{(j)}+(h) - b^{(j)}+(h - 1) \). Note that (A.54) is still true at \( k = 0 \), with the convention \( b^{(j)}-(0) = 0 \). We are going to prove (A.37) by induction with \( q \in \{1, ..., n\} \).

Applying formula (A.36) of fact A.11 at \( q = n \), we have
\[ \forall h = 1, ..., l, \ y(k + h) = \sum_{j=0}^{n-1} S_j(h - j) y^{(j)}(k + 1) \]

Using (A.54) with \( j = n - 1 \) and the fact that \( y = b^{(n)}- \), we find that, for \( h \in \{1, ..., l\} \),
\[ b^{(n-1)}-(k + h) = b^{(n-1)}-(k) + \sum_{r=1}^{h} y(k + r) \]
\[ = b^{(n-1)}-(k) + \sum_{r=1}^{h} \left( \sum_{j=0}^{n-1} S_j(r - j) y^{(j)}(k + 1) \right) \]
\[ = b^{(n-1)}-(k) + \sum_{j=0}^{n-1} \left( \sum_{r=1}^{h} S_j(r - j) \right) y^{(j)}(k + 1) \]

Using fact A.9 and the fact that \( S_0(h) = 1 \) we have
\[ b^{(n-1)}-(k + h) = S_0(h) b^{(n-1)}-(k) + \sum_{j=0}^{n-1} S_{j+1}(h - j) y^{(j)}(k + 1) \]

This proves (A.37) for \( q = 1 \).

Now, let us suppose that (A.37) is true for a certain \( q \in \{1, ..., n-1\} \). Let us consider \( h \in \{1, ..., l\} \). Applying (A.54) with \( j = n - q - 1 \), we find
\[ b^{(n-q-1)}-(k + h) = b^{(n-q-1)}-(k) + \sum_{r=1}^{h} b^{(n-q)}-(k + r) \]

Using the assumption that fact A.12 is true at \( q \), we can write
\[ \sum_{r=1}^{h} b^{(n-q)}-(k + r) = \sum_{r=1}^{h} \left( \sum_{j=0}^{q-1} S_j(r) b^{(n-q+j)}-(k) + \sum_{j=0}^{n-1} S_{q+j}(r - j) y^{(j)}(k + 1) \right) \]
\[ = \sum_{j=0}^{q-1} \left( \sum_{r=1}^{h} S_j(r) b^{(n-q+j)}-(k) \right) + \sum_{j=0}^{n-1} \left( \sum_{r=1}^{h} S_{q+j}(r - j) \right) y^{(j)}(k + 1) \]
Using fact A.9, performing the change of variable $j' = j + 1$ in the first summation, and using the fact that $S_0(h) = 1$, (A.55) becomes

$$\forall h \in \{1,...,l\}, \quad b^{(n-(q+1))}(k + h) = \sum_{j'=0}^{(q+1)-1} S_{j'}(h)b^{(n-(q+1)+j')}(k) + \sum_{j=0}^{n-1} S_{(q+1)+j}(h - j)y^{(j)}(k + 1)$$

This proves (A.37) for $(q + 1)$. The induction is completed $\Box$

A.5 Proof of fact 4.14

We first have the following lemma

**Lemma A.13** A signal $b$ minimizes $\phi$ subject to $\mathcal{E}(I_p)$ if and only if

$$\forall i = 1,...,p-1 \quad , \quad b(k_i) = 0 \quad \text{and} \quad b(k_p) = 1 \quad \text{(A.56)}$$

$$\forall q = 0,...,n-1 \quad , \quad y^{(q)}(k_p + 1) = 0 \quad \text{(A.57)}$$

$$\forall k \notin I_p \quad , \quad y^{(n)}(k) = 0 \quad \text{(A.58)}$$

where $y = b^{(n)-}$.

Proof: ($\Rightarrow$) If $b$ minimizes $\phi$ subject to $\mathcal{E}(I_p)$, then $b = b_p$. Relation (A.56) is trivial and (A.58) is a consequence of theorem 4.3 applied to $S = \mathcal{E}(I_p)$. Property (A.57) comes from fact 4.9.

($\Leftarrow$) We show that a sequence satisfying (A.56), (A.57) and (A.58) verifies the Euler inequality applied to $S = \mathcal{E}(I_p)$. First, (A.56) directly implies that $b \in \mathcal{E}(I_p)$. Let us take any $b' \in \mathcal{E}(I_p)$. Using (A.56), (A.58) and the fact that $\frac{\partial \phi}{\partial b} = (-1)^n y^{(n)}$ where $y = b^{(n)-}$ (from fact 4.5), we have

$$\sum_k \frac{\partial \phi}{\partial b}(k) \cdot (b'(k) - b(k)) \geq 0 \quad \square$$

Since $\forall k \notin I_p$, $y^{(n)}(k) = 0$, we have, for every $i = 1,...,p$,

$$\forall l = 1,...,l_i - 1, \quad y^{(n)}(k_{i-1} + l) = 0$$

Applying fact 4.12 on $b = b_p$ with $k = k_{i-1}$, $l = l_i$, we find that

$$U_{b_p}(k_i) = M(l_i)U_{b_p}(k_{i-1}) + N(l_i)(b_p(k_i) - b_p(k_{i-1}))$$

Since $b_p(k_i) = 0$ for $i = 0,...,p-1$ and $b_p(k_p) = 1$, we have

$$U_{b_p}(k_i) = M(l_i)U_{b_p}(k_{i-1}) \quad \text{for} \quad i = 1,...,p-1$$

$$U_{b_p}(k_p) = M(l_p)U_{b_p}(k_{p-1}) + N(l_p) \quad \text{for} \quad i = p$$

If we define for $i = 1,...,p$, $M_i = M(l_i)M(l_{i-1})...M(l_i)$, then

$$U_{b_p}(k_i) = M_iU_{b_p}(0) \quad \text{for} \quad i = 1,...,p-1$$

(A.59)
\[ U_{b_i}(k_p) = M(l_p)U_{b_i}(k_{p-1}) + N(l_p) \text{ for } i = p \]  \hspace{1cm} (A.60)

For \( i = 0, ..., p \), we have
\[ W_{p,i}^0 = P U_{b_i}(k_i) \]  \hspace{1cm} (A.61)

and in the particular case of \( i = 0 \) \( (k_0 = 0) \), we have
\[ U_{b_0}(0) = P T W_{p,0}^0(0) \]  \hspace{1cm} (A.62)

Relations (A.59), (A.61) and (A.62) imply that
\[ \forall i = 1, ..., p - 1 \text{ , } W_{p,i}^0 = (P M_i P^T) W_{p,0}^0 \]  \hspace{1cm} (A.63)

Using the fact that \( W_{p,p}^0 = O_{n-1} \) (consequence of (A.57)), relations (A.60), (A.61) and (A.62) imply that
\[ 0 = (P M_p P^T) W_{p,0}^0 + P N(l_p) \]

If \( (P M_p P^T) \) was not invertible, it would be possible to find a vector \( W_{p,0}^0 \neq W_{p,0}^0 \) such that
\[ 0 = (P M_p P^T) W_{p,0}^0 + P N(l_p) \]

One can check that it would be possible to construct a sequence \( b_p' \neq b_p \) which satisfies (A.56), (A.57) and (A.58): this is absurd by unicity of the minimum. Therefore, \( (P M_p P^T) \) is necessarily invertible. Then,
\[ W_{p,0}^0 = -(P M_p P^T)^{-1} P N(l_p) \]

Relation (A.63) becomes
\[ \forall i = 1, ..., p - 1 \text{ , } W_{p,i}^0 = -(P M_i P^T)(P M_p P^T)^{-1} P N(l_p) \]  \hspace{1cm} (A.64)

From the recursive definition of \( R_i \) and \( M_i \) it is easy to verify that
\[ R_p = M_p P^T \text{ and } M_p = \begin{bmatrix} P P^T \\ P M_1 P^T \\ \vdots \\ P M_{p-1} P^T \end{bmatrix} \]

We know that \( PR_p = P M_p P^T \) is invertible. Therefore (A.64) implies formula (4.9). Next, we have
\[ V_p = Q U_{b_p}(k_p) \]

Applying (A.60), (A.62) and using the fact that \( R_p = M_p P^T \), we find formula (4.10) \( \square \)

72
A.6 Study of algorithm 3.4 in the case \( n = 2 \)

A.6.1 Expression of \( W_{p,i}^0, V_p^0 \)

From appendix A.4.3, the expressions of \( M(l) \) and \( N(l) \) are

\[
M(l) = -\frac{1}{l^2 - 1} \begin{bmatrix} 2l^2 + 3l + 1 & 6l \\ l^2(l^2 - 1) & 2l^2 + 3l + 1 \end{bmatrix} \quad \text{and} \quad N(l) = \frac{3}{l^2 - 1} \begin{bmatrix} 2 \\ l - 1 \end{bmatrix}
\]

We have \( P = [1 \ 0] \) and \( Q = [0 \ 1] \). Matrix \( R_p \) is just a column vector of size 2. Let us write \( R_p = \begin{bmatrix} \alpha_p \\ \beta_p \end{bmatrix} \). Then \( PR_p = \alpha_p \) and \( QR_p = \beta_p \). The recursive construction (4.8) becomes

\[
\alpha_0 = 1, \ \beta_0 = 0, \ \mathcal{M}_0 = [] \quad \text{and} \quad \forall i \geq 1, \quad \begin{bmatrix} \alpha_i \\ \beta_i \end{bmatrix} = M(l_i) \begin{bmatrix} \alpha_{i-1} \\ \beta_{i-1} \end{bmatrix}, \quad \mathcal{M}_i = \begin{bmatrix} \mathcal{M}_{i-1} \\ \alpha_{i-1} \beta_{i-1} \end{bmatrix}
\]

Consequently \( \mathcal{M}_p = \begin{bmatrix} \alpha_0 \\ \vdots \\ \alpha_{p-1} \end{bmatrix} \). From (4.9) and (4.10) we find

\[
W_{p,i}^0 = -\frac{6}{l_p-1} \frac{\alpha_i}{\alpha_p} \quad \text{for} \quad i = 0, \ldots, n - 1 \quad \text{and} \quad V_p^0 = \frac{2}{l_p-1} \left( l_p - 1 - 2\beta_p \right)
\]

A.6.2 Proof of conjecture 4.6 in the case \( n = 2 \)

By assumption of conjecture 4.6, \( \forall i = 1, \ldots, p, l_i \geq 2 \). When \( l \geq 2 \), the coefficients of \( M(l) \) are always negative. This implies that \( \alpha_1 \) and \( \beta_1 \) are negative since

\[
\begin{bmatrix} \alpha_1 \\ \beta_1 \end{bmatrix} = M(l_1) \begin{bmatrix} 1 \\ 0 \end{bmatrix}.
\]

From (A.65), it is easy to see by induction that \( \forall i \geq 1, \ sign(\alpha_i) = sign(\beta_i) = (-1)^i \). This implies that \( sign(W_{p,i}^0) = (-1)^{p+1-i} \). In the context \( n = 2, W_{p,i}^0 = y_p(k_i+1) \). We know from fact 4.10 that \( y_p = y_p^{(n-2)+} \) is piecewise linear on intervals of the type \( \{k_{i-1} + 1, \ldots, k_i + 1\} \).

On such an interval the slope \( (y_p(k_{i+1}) - y_p(k_{i-1}))/l_i \) of \( l_i \) is necessarily of sign \( (-1)^{p+1-i} \). One can verify that this is also true at \( i = p \), using the fact that \( y_p(k_p+1) = 0 \) (fact 4.9). This implies that the variation of slope of \( y_p \) about time index \( k_i + 1 \) has a sign equal to \( (-1)^{p-i} \) for \( i = 1, \ldots, p - 1 \). One can check that this is also true for \( i = p \) since the slope of \( y_p \) becomes 0 from \( k \geq k_p + 1 \) (fact 4.9). Since

\[
y_p^{(2)+}(k) = (y_p(k+2) - y_p(k+1) - (y_p(k+1) - y_p(k))
\]

is the variation of slope of \( y_p \) about \( k + 1 \), we have just shown that

\[
\forall i = 1, \ldots, p, \ sign\left(y_p^{(2)+}(k_i)\right) = (-1)^{p-i} \quad \text{This leads to (4.4) since} \quad y^{(2)+} = \frac{\partial \alpha}{\partial y_p} \quad \text{from fact 4.5}
\]

A.6.3 Exponential decay of \( W_{p,i}^0 \) for increasing \( p \)

Since \( \alpha_i \) and \( \beta_i \) have always the same sign and \( M(l) \) have all negative coefficients, from the recursive relation (A.65) we can write

\[
|\alpha_i| = \frac{1}{l_i-1} \left((2l_i^2 + 3l_i + 1)|\alpha_{i-1}| + 6l_i|\beta_{i-1}| \right) \geq \frac{2l_i^2 + 3l_i + 1}{l_i-1} |\alpha_{i-1}|
\]

73
From algorithm 3.1, \( l_i \) is always chosen greater than 2. Since \( \frac{2^i+2^{i-1}}{2^{i-1}} = \frac{2^i+1}{i+1} \) is an increasing function, its minimum for \( i \geq 2 \) is reached at \( i = 2 \) and is worth \( \frac{5}{2} \). Therefore, \( |\alpha_i| \geq \frac{5}{2}|\alpha_{i-1}| \) which means that \( \alpha_i \) increases exponentially. This implies that for a fixed \( i \),
\[
|W_{p+i}^0| = \frac{6}{p-1} |\alpha_i| |\alpha_p| \leq 2 |\alpha_i| |\alpha_p|
\]
decays exponentially when \( p \) increases.

### A.7 Proof of fact 4.15

Let \( (R_t)_{t \leq i \leq p} \) and \( (\mathcal{M}_t)_{t \leq i \leq p} \) be the matrix sequences recursively defined by (4.8), \( (T_i)_{t \leq i \leq p} \) be an arbitrary sequence of invertible matrices of size \( (n-1) \), \( (R'_t)_{t \leq i \leq p} \) and \( (\mathcal{M}'_t)_{t \leq i \leq p} \) the matrix sequences recursively defined by

\[
\begin{align*}
R_0' &= P^T \\
\mathcal{M}_0' &= [ ] \\
\text{and } \forall i = 1, \ldots, p, \quad R_i' &= M(i) R_{i-1}' R_i \\
\mathcal{M}_i' &= \left[ \mathcal{M}'_{i-1} \right] T_i 
\end{align*}
\]

(A.66)

For every \( p \geq 0 \), we define \( S_p = T_1 T_2 \ldots T_p \) \( (S_0 = I_{n-1,n-1}) \). Let us show by induction the proposition

\( Q(p) : \quad \mathcal{M}_p' = \mathcal{M}_p S_p \quad \text{and} \quad R_p' = R_p S_p \)

We have: \( \forall p \geq 0, S_p T_{p+1} = S_{p+1} \).

\( Q(0) \) is true because \( S_0 = I_{n-1,n-1}, \mathcal{M}_0 = \mathcal{M}_0' = [ ] \) and \( R_0 = R_0' = P^T \).

Now, suppose that \( Q(p) \) has been proved for some \( p \geq 1 \). Then

\[
\mathcal{M}_{p+1}' = \begin{bmatrix} \mathcal{M}_p' \\ PR_p' \end{bmatrix} T_{p+1} = \begin{bmatrix} \mathcal{M}_p S_p \\ PR_p S_p \end{bmatrix} T_{p+1} = \begin{bmatrix} \mathcal{M}_p \\ PR_p \end{bmatrix} S_{p+1} = \mathcal{M}_{p+1} S_{p+1}
\]

and

\[
R_{p+1}' = M(K_p) R_p' T_{p+1} = M(K_p) R_p S_p T_{p+1} = M(K_p) R_p S_{p+1} = R_{p+1} S_{p+1}
\]

This implies that \( Q(p+1) \) is true. The induction is completed.

Applying \( Q(p) \) and the fact that \( S_p \) is invertible, we have

\[
(QR_p')(PR_p')^{-1} = (QR_p')(PR_p)^{-1} \quad \text{and} \quad \mathcal{M}_p'(PR_p')^{-1} = \mathcal{M}_p(PR_p)^{-1}
\]

We finally complete the proof by using these equalities in (4.9) and (4.10) respectively \( \square \)
Bibliography


