Télécom ParisTech
Département TSI

Convex optimization for signal reconstruction from short-term Fourier transform magnitude

Master ATIAM
internship report

Author:
Guillaume Mahenc
guillaume_mahenc@hotmail.fr

Supervisors:
Matthieu Kowalski
Hélène Papadopoulos
Gaël Richard

July 31, 2013

This report sums up a Master of Science internship done between March and July 2013 at Télécom Paris. The original purpose was to test the convex method for signal reconstruction from STFT magnitude proposed by Sun and Smith in their paper of 2012 [Sun and Smith, 2012] and to compare it to the state of the art, in particular the now standard Griffin-Lim algorithm [Griffin and Lim, 1984]. The Sun-Smith algorithm performs well on very small sizes of signal. From 128 samples it becomes very slow because of the multiple diagonalizations that are very computational. It was then decided to try to improve the Griffin-Lim algorithm. Exploiting the mathematical properties of the short-term Fourier transform, in particular the variation of magnitude under temporal phase shifts, several questions arose. Most of them, unanswered, are stated at the end of this report.

Key-words: short-term Fourier transfom, magnitude, signal reconstruction, phase, phase retrieval, convex optimization, semi-definite programming, relaxation, PhaseLift, Hilbert transform, time-frequency representation.

Ce rapport résume l'ensemble des travaux de stage de M2 effectués par l'auteur entre mars et juillet 2013 à Paris dans les locaux de Télécom. L'objectif originel de ce stage était de comparer la récente méthode convexe proposée par Sun et Smith [Sun and Smith, 2012] dans leur article de 2012 pour reconstruire un signal temporel à partir du module de sa transformée de Fourier à court terme, puis de la comparer à l'état de l'art, en particulier au classique algorithme de Griffin et Lim [Griffin and Lim, 1984]. L'algorithme de Sun et Smith s'est avéré performant pour des signaux de très courte durée. Au-delà de 128 échantillons cependant, les diagonalisations successives le rendent très lent. Il a par la suite été décidé de chercher par diverses manières d'améliorer l'algorithme de Griffin et Lim. Pour cela, les propriétés mathématiques de la transformée de Fourier à court-terme ont été explorées, en particulier la variation de module causée par un déphasage temporel. Le recoupement des résultats théoriques avec des simulations numériques et avec les résultats que donne l'algorithme de Griffin et Lim ont été la source de nombreuses questions dont la plupart sont pour l'instant restées sans réponse.

Mots-clés : transformée de Fourier à court terme, module, reconstruction de signal, phase, reconstruction de phase, optimisation convexe, programmation semi-définie, relaxation, PhaseLift, transformée de Hilbert, représentation temps-fréquence.

Acknowledgements

I would like to thank Gaël Richard for his precious advice on the signal processing part of the work, Matthieu Kowalski for the personal courses on convex optimization, and Hélène Papadopoulos for her advice on the redaction of the report. Thank you for having allowed me to do this internship.

I would like to thank Dennis Sun who kindly accepted to send me his Matlab code and therefore made my life much easier.

I would like to thank Katie Smith for her advice on how to properly write in English, even though I still have to improve my skills.

I also would like to thank Simon Durand for having been both a coworker and a musical partner, and François Rigaud and Mounira Maazaoui for the stoner rock part of this internship.

Contents

Notations

This page sums up the notations used in the report. These notations do not hold in a section where another use is made of the symbols.

Indices, vectors, matrices

- k, n, m refer to the index of a discrete signal.
- $0 : N-1$ refers to the set $\{0, \cdots, N-1\}$. It is generally used in order to define the support vector of a discrete function. Example: $\sin(2\pi(0 : N-1)) = \{\sin(2\pi n), n \in \{0, \cdots, N-1\}$. This is actually a Matlab-like notation.
- t, τ refer to a continuous time variable.
- ω refers to a continuous frequency variable.
- x, y, s generally refer to temporal signals. If it is not precised if they are discrete or continuous, the index may help.
- When there is no ambiguity on a temporal signal, F refers to its Fourier transform.
- When there is no ambiguity on a temporal signal, S refers to its short-term Fourier transform.
- When there may be an ambiguity, if x_1 (resp. x_2) is a temporal signal, X_1 (resp. X_2) refers to its short-term Fourier transform.

Conjugation, scalar products, norms

- i and j refer to the same unitary imaginary number defined by $i^2 = j^2 = -1$.

Let z be a complex number.

- $\Re$ (resp. $\Im$) refer to its real (resp. imaginary) part, so that z can be written as $z = \Re(z) + i\Im(z)$ with $(\Re(z), \Im(z)) \in \mathbb{R}^2$.
- z^* refers to the complex conjugate of z.
- $|z| = \sqrt{zz^*}$ refers to the magnitude of z. The magnitude is a 2-norm on complex numbers.

Complex-valued functions

Let f be a complex-valued function, discrete or continuous. Note that a discrete function with finite support can be represented as a vector in a finite dimension vector space. Example: $f(0 : L-1) \in \mathbb{C}^L$ means that f has for support $0 : L-1$.

f^* refers to the complex conjugate of f, i.e. to the complex function $f = \Re(f) - i\Im(f)$.

If f and g are two complex-valued functions defined on a same set $\mathcal{D} \subset \mathbb{R}^N$, the scalar product of f and g is defined as

$$\langle f, g\rangle = \int_{\mathcal{D}} f(t_1, \ldots, t_N) g^*(t_1, \ldots, t_N) dt_1 \ldots dt_N.$$

The L^2-norm of f is noted $\|f\|_2$ and defined by

$$\|f\|_2^2 = \langle f, f\rangle = \int_{\mathcal{D}} |f(t_1, \ldots, t_N)|^2 dt_1 \ldots dt_N.$$

Complex matrices

Let A be a $M \times N$ matrix.

$$A = \begin{pmatrix} a_{0,0} & \cdots & a_{0,N-1} \\ \vdots & & \vdots \\ a_{M-1,0} & \cdots & a_{M-1,N-1} \end{pmatrix}$$

A^* refers to its $N \times M$ conjugate transpose. Note that a vector is a matrix with only one line or column.

$$A^* = \begin{pmatrix} a^*_{0,0} & \cdots & a^*_{M-1,0} \\ \vdots & & \vdots \\ a^*_{0,N-1} & \cdots & a^*_{N-1,M-1} \end{pmatrix}$$

If A is squared, i.e. $M = N$, its trace is defined by $\mathbf{tr}(A) = \sum_{i=0}^{N-1} a_{i,i}$.
If A and B have the same size, the usual scalar product is

$$\langle A, B\rangle = \mathbf{tr}(AB^*).$$

It defines the Fröbenius norm $\|\cdot\|$ such that $\|A\|^2 = \mathbf{tr}(AA^*)$.

Transforms

In this report, short-term Fourier transform, Gabor transform and spectrogram refer to the same concept of mapping a signal depending only on time towards a signal depending both on time and on frequency.

- $\mathcal{F}$ refers to the Fourier transform operator. If x is discrete (resp. continuous), $F = \mathcal{F}(x)$ is its discrete- (resp. continuous-) time Fourier transform.
- $\mathcal{S}$ refers to the short-term Fourier transform operator. In the discrete case, it maps a vector x to a matrix $S = \mathcal{S}(x)$ being its short-term Fourier transform. In the continuous case, it maps a one-temporal-variable function $x(t)$ to a two-variable function $S(\tau, \omega)$, one variable being temporal (τ) and the other one being a frequency variable (ω).
- Ψ refers to the Gabor transform. The difference between Ψ^* and $\mathcal{S}$ is conceptual.
- $\mathcal{H}$ refers to the Hilbert transform.
- The identity operator or matrix is noted I or Id.

Télécom ParisTech - Département TSI

The TSI department is one of the four departments in charge of teaching and research within Télécom ParisTech. The acronym TSI stands for "Traitement du Signal et des Images" meaning Signal and Image Processing. For its research activities, the department belongs to UMR CNRS 5141 LTCI.

The missions of TSI are teaching and research in the domain of signal processing and image processing. Its objectives are:

- studies on images under all its forms, digital, optical... for various applications : medical, satellite, artistic...,
- studies on speech, sound and music,
- studies on multimedia documents, video.

The main research topics are:

- development of algorithms and statistical processing techniques, in particular for model learning,
- multimedia indexation, sensor networks, biometrics,
- coding and transmission for multimedia communications.

Information taken from the website `http://www.tsi.telecom-paristech.fr/`

Introduction: The phase recovery problem

The challenges of phase recovery

This section presents the main challenges of phase recovery and the different fields of application, from 1-D signal to 2-D image signal processing or, of course, STFT magnitude.

Finding a signal from Fourier data with missing phase information has been a challenge for at least half a century. Some measurements in physics are made in the Fourier domain and only the magnitude can be actually measured. In order to get the associated temporal signal, one has to find the phase from the magnitude data.

In his 1978 PhD thesis on time-scale modification of speech based on short-time Fourier analysis [Portnoff, 1978], Michael R. Portnoff finding an "improved estimate of the fm component of the phase of the short-time Fourier transform, perhaps, by a better estimator than [a] simple difference scheme" as further research. Here is another reason why we should do phase retrieval, and which is more closely related to music or speech. Some modifications of a sound signal are made on their short-term Fourier transform and only operate on its magnitude, while the phase carries audible information. That is why we shall focus, in this study, on the reconstruction of an audio signal from the magnitude of its short-term Fourier transform.

The general phase recovery problem

The phase recovery problem can be formulated in a general way as suggested by [Waldspurger et al., 2012]. Let $x \in \mathbb{C}^p$ denote the signal we want to retrieve, but assume we only know the amplitude $b = |Ax| \in \mathbb{R}^n$ of n linear measurements. A denotes a linear operator, i.e. a matrix of $\mathbb{C}^{n\times p}$ which can be a STFT operator in sound processing, or a 2-D Fourier operator in image processing. The phase retrieval problem can be formulated as

$$\begin{array}{rl} \text{find} & x \\ \text{such that} & |Ax| = b \end{array} \tag{1}$$

where the operator $|\cdot|$ on $\mathbb{C}$ is such that $|y| = b \Leftrightarrow |y_i| = b_i,\ i = 1, \ldots, n$.

The main existing algorithms

The main algorithms of the literature are presented, essentially [Gerchberg and Saxton, 1971] and [Griffin and Lim, 1984]. They are all alterned projections algorithms.

The Gerchberg-Saxton algorithm

The Gerchberg-Saxton "puts forward a rapid computational method for determining the complete wave function (amplitudes and phases) from intensity recordings in the image and diffraction planes" [Gerchberg and Saxton, 1971]. Published in the journal *Optik* in 1971, it was to be applied to 2-D signals of image recordings. It uses mathematical properties of the 2-D Fourier transform: "the method depends on there being a Fourier Transform relation between the waves in these two planes and hence constrains the degree of temporal and/or spatial coherency of the wave."

Actually this is an alterning projection algorithm. It computes iterates y^1, y^2, $\cdots$ of a spectral signal y whose only magnitude is known. The first projection step is the application $AA^\dagger$. The second step is the projection onto the set of complex matrices with same magnitude. Waldspurger et al. sum up the Gerchberg-Saxton algorithm in [Waldspurger et al., 2012]. They call

$$\mathbb{F} = \{y \in \mathbb{C}^n, |y| = b\}$$

Data: An initial $y^1 \in \mathbb{F}$
Result: $y^N \in \mathbb{F}$
for $k = 1, \ldots, N-1$ **do**

$$y_i^{k+1} = b_i \frac{(AA^\dagger y^k)_i}{|(AA^\dagger y^k)_i|}, \; i = 1, \ldots, n$$

end

Algorithm 1: The Gerchberg-Saxton algorithm as explained by Waldspurger

The Griffin-Lim algorithm

The Griffin-Lim algorithm has been the most widely used phase-retrieval algorithm for audio signals for the last 30 years.

It is a reformulation of the Gerchberg-Saxton algorithm to the case where instead of the operator A is a short-time Fourier transform operator Ψ, such that $\Psi^\dagger x(\tau, \omega)$ is the short-time Fourier transform of x. The Griffin-Lim algorithm can be summed up as follows. L is the length of the temporal signal s. M and N are respectively the number of rows and the number of lines of the STFT matrix. Those parameters will be explained in the chapter on time-frequency analysis. S^k denotes the STFT calculated by the algorithm at k^{th} iteration of the algorithm. The last iterate called I_{max} generally depends on the convergence of the distance measure between estimate and objective on STFT magnitude.

Data: An initial STFT magnitude $W \in \mathbb{R}_+^{MN}$
Result: $s \in \mathbb{R}^L$
while $k < I_{max}$ **do**

$$S^{k+1}(\tau, \omega) = W(\tau, \omega) \frac{\Psi\Psi^\dagger S^k(\tau, \omega)}{|\Psi\Psi^\dagger S^k(\tau, \omega)|}$$

end

$$s = \Psi S^{I_{max}}$$

Algorithm 2: The Griffin-Lim algorithm

The Griffin-Lim algorithm has become a classic method for short-time Fourier phase retrieval. It has been shown since its creation that it is not convex. Sturmel and Daudet clearly explain in [Sturmel and Daudet, 2011] what problems the Griffin-Lim algorithm may encounter. In particular, stagnation is caused by an indetermination of the phase difference from one windowed section to another. It may lead to sudden sign inversions in the reconstructed signal, as shown in figure 1. Such a signal is a stable local minimum for the distance used as convergence criterion. The development of convex optimization and the applications to signal processing, in particular in image processing and X-ray cristallography, led to the attempt of find a convex method for short-time Fourier phase retrieval.

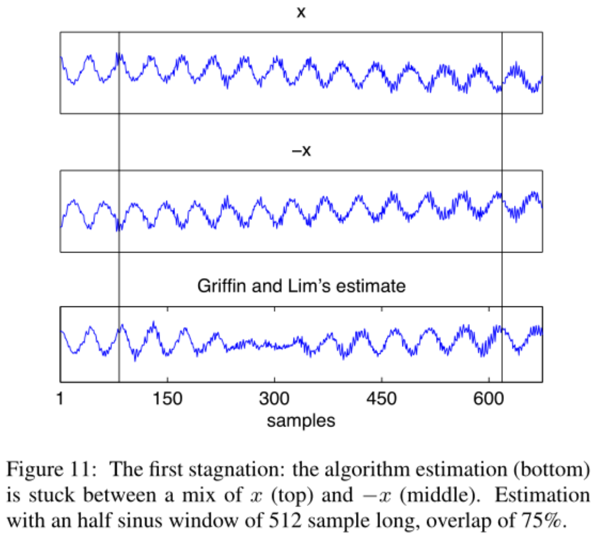

Figure 1: Example of stagnation taken from [Sturmel and Daudet, 2011]

Convex methods

Recently, in 2012, Dennis Sun, who is currently doing his PhD at Stanford, adapted techniques of convex optimization developed by Candès [Candès et al., 2011] to the case of the STFT. He thus presents a method called STlifT [Sun and Smith, 2012] that converges to a unique solution, claiming that it is a more efficient method than the previous non convex ones.

Convex optimization is quite a recent theory in programming, it is a generalization of linear programming. Once again it is an alterned projection method.

One has to define a cost function that must be convex and constraints that must be convex too. Those constraints are to reduce the set of definition to a smaller subset where the solution is likely to be found. If both the cost function and the constraint are convex, different theorems allow to build a converging sequence of elements towards the minimum argument of the cost function. That minimum is unique thanks to convexity.

In order to retrieve the phase of the STFT using convex optimization, we are going to use a projected gradient method. A condition for applying such a method is that the gradient of the cost function is Lipschitzian. The cost function does not have to be differentiable all over the set of definition. The projected gradient method makes use of the proximal operator associated to the cost function.

Objectives

This report sums up the work done during a 4.5 month internship. Several problematics were raised concerning the phase retrieval of the short-time Fourier transform. The most important, although very simple, question that may sum up those problematics can be formulated as :

How does the redundancy of information in short-term Fourier transform make possible that part of the phase information is contained in the magnitude data?

This report does not bring the definitive answer to that question, as the objective of science is more to understand the questions than to find the answers. Theroetical considerations both on continuous signals - for convenience of the calculus - and on discrete signals - for the implementation purposes that are inherent to any knid of research on discrete signal processing - try to find out where the phase information that is lost *a priori* can be extracted.

The most important elements of answer are given by the algorithms that build a phase from the magnitude. However, although the convergence of the algorithms is mathematically proven, a better understanding of their framework could help to improve them.

Organization of the report

The first chapter of this report brings a short mathematical overview of what is nowadays called the *phase*, especially in the signal processing field. The notion of phase of a discrete real signal is not very well defined, although the use of the Hilbert transform could help figuring out what it could be.

The second chapter starts where the first ended, that is on the time-frequency analysis. This discipline is not a hundred years old yet, but has been, under a different name, the theoretical background of the major scientifical and philospohical breakthrough of the twentieth century : quantum mechanics. The present report does not extend on the links between signal processing and quantum mechanics, although a long and extremely interesting discussion could be possible. It focuses on the different interpretations of the time-frequency representations that are associated to different streams of research of the 20th and 21st centurt, mainly:

- Gabor pioneering, a communication theory approach,
- the renewal in the 1970's due to the phase vocoder,
- the latest algebraic approach led by wavelet theory.

The third and fourth chapters present the algorithms that were studied and implemented during the internship. The Griffin-Lim algrorithm, that is older, is presented first, with the difficulties of reconstruction that arise from its implementation. Then, an amelioration of the algorithm is discussed, made on assumption that are justified both by calculus and numerical simulations.

The convex optimization method is presented in the fourth chapter. It sums up the recent article by Dennis Sun and Julius Smith that proposes another algorithm. The main difficulty of this algorithm is that it needs diagonalize a very large matrix.

In the fifth chapter, the Griffin-Lim algorithm and Sun-Smith (STliFT) algorithms are compared on two types of signals :

- white noise random signals,
- pure sine signals,

taking into account both the precision of the reconstruction and the time of calculation.

Chapter 1

Time-Frequency Analysis

This chapter presents the main results in time-frequency analysis.

1.1 Gabor pioneering

The physicist Dennis Gabor (1900-1979), Nobel Prize in 1971, makes a description of the goals of time-frequency analysis in his *Theory of Communication* [Gabor, 1946]. Essentially focused on the "transmission of data", he presents "points on which physical feeling and the usual Fourier methods are not in perfect agreement". As an example, working on complex signals is the only way to have only positive frequency components.

Gabor was interested in different ways to represent signals. He defines an "elementary signal" as

$$\psi(t) = \exp\left(-\alpha^2(t-t_0)^2 + j(2\pi f_0 t + \phi)\right)$$

that is a pure complex monochromatic signal windowed in time by a gaussian. This signal has gaussian envelopes both in time and frequency with respective temporal and spectral width

$$\begin{cases} \Delta t = \sqrt{\dfrac{\pi}{2}}\dfrac{1}{\alpha} \\ \Delta f = \dfrac{1}{\sqrt{2\pi}}\alpha \end{cases}$$

that ensures that the Heisenberg inequality is here an equality:

$$\Delta t \Delta f = \frac{1}{2}.$$

A signal like ψ ensures that each rectangle of size $\Delta t \times \Delta f$ in the time-frequency plane has the minimal size allowed by the Heisenberg inequality. For Gabor, an interpretation on data transmission is that such a rectangle carries the minimal amount of information. He calls the information a *logon*. For a given signal, the corresponding logon are complex values that can be stocked into a time-frequency matrix $C = (c_{nk})$, each coordinate of the matrix corresponding to its coordinate in the time-frequency plane, discretised such that

$$\begin{cases} t_n = n\Delta t \\ f_k = k\Delta f \end{cases}$$

An example of such a matrix, that can be infinite for unbounded signals, is presented on figure 1.1.

Gabor wanted to represent any kind of signal as a weighted sum of signals, given that one can switch from the matrix representation (c_{nk}) to a temporal representation $\psi(t)$ by summing over all time frames n and frequency frames k.

$$\psi(t) = \sum_{n=-\infty}^{\infty}\sum_{k=-\infty}^{\infty} c_{nk} \exp\left(-\pi\frac{(t-n\Delta t)^2}{2(\Delta t)^2} + j2\pi\frac{kt}{\Delta t}\right)$$

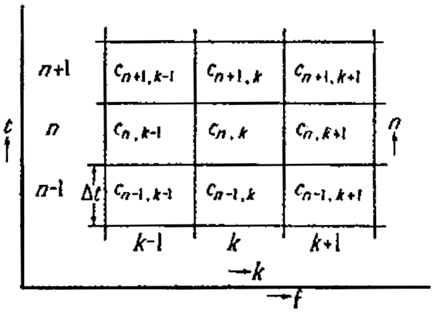

Figure 1.1: A representation of signal by a matrix of complex amplitudes proposed by Gabor [Gabor, 1946]

One can see in this formula the expression of the discrete inverse short-time Fourier transform with a Gaussian window. This is probably one of the first attempts to build a time-frequency representation. It took place in 1946. It was then established that there are more accurate ways to represent a signal in the time-frequency plane, using different windows of analysis like Hann, Hamming or Blackman. The main reason to use a different analysis and/or synthesis window in Fourier analysis is its spectral properties, like the bandwidth of the main lobe and the decreasing rate of the sidelobes. This properties are very important when working on discrete signals because spectral leakage due to the use of a discrete transform is unavoidable.

1.2 A unified approach in the 1970s

The use of the short-term Fourier transform became common in the 1970s, when computers became powerful enough to compute Fourier transform in reasonnable amounts of time – and when Cooley and Tukey successfully introduced their algorithm of fast Fourier transform [Cooley and Tukey, 1965]. The work of Flanagan [Flanagan and Golden, 1966] and Portnoff [Portnoff, 1976] on the phase vocoder made a use of the short-term Fourier transform. Different interpretations and techniques were formulated concerning the STFT and its inversion. This multiplicity led the researchers Jont B. Allen and Lawrence R. Rabiner to publish *A Unified Approach to Short-Time Fourier Analysis and Synthesis* in 1977 [Allen and Rabiner, 1977]. They sum up the main definitions and results useful for time-frequency analysis. Of course, this list has been widely extended since the wavelet approach, but it remains a useful toolbox for understanding the framework of STFT.

1.2.1 Definition of the short-time Fourier transform for discrete signals

They give the following definition of the STFT of a discrete real signal $x(m), m \in \mathbb{Z}$ using an analysis window $w(m)$:

$$X_n(e^{j\omega_k}) = \sum_{m=-\infty}^{\infty} w(n-m)x(m)e^{-j\omega_k m}, n, k \in \mathbb{Z}.$$

Note that $X_n(e^{j\omega_k})$ can be seen as the $(n,k)^{th}$ element of a matrix, just like the (c_{nk}) coefficients we were previously talking about. n stands for a discrete time frame and $\omega_k \in [0, \pi]$ for a discrete frequency variable, up to multiplication by 2π. Note that due to Hermitian symmetry we do not need to consider the STFT on negative frequencies.

One of the main difficulties in time-frequency analysis is the multiplicity of the interpretations of the transforms.

Interpretation as a convolution product

Allen and Rabiner give a first interpretation of the STFT as a convolution product:

$$X_n(e^{j\omega_k}) = [x(n)e^{-j\omega_k n}] * w(n).$$

Here $X_n(e^{j\omega_k})$ is viewed as a function of n for fixed ω_k.

Interpretation as a simple Fourier transform

The second interpretation consists of seeing $X_n(e^{j\omega_k})$ as a function of ω_k for fixed n. Let $y_n(m)$ be the signal $x(m)$ windowed by the reversed analysis window $w(-m)$ shifted of n samples:

$$y_n(m) = x(m)w(n-m).$$

Then the STFT of $x(m)$ on frame s is the regular Fourier transform of y_n:

$$X_n(e^{j\omega_k}) = \mathcal{F}\{y_n\}(e^{j\omega_k}), n, k \in \mathbb{Z}. \tag{1.1}$$

1.2.2 Inversion of the short-time Fourier transform

As there were two ways to interpret the short-time Fourier transform, two methods exist to built the temporal signal $x(m)$ given the comples coefficients $X_n(e^{j\omega_k})$ of its STFT. As Gabor explained in [Gabor, 1946], the reconstruction is made by summing over all the temporal and frequency coefficients. Here again, we present the article of Allen and Rabiner [Allen and Rabiner, 1977].

Filter-Bank Summation (FBS)

This method uses the assumption that the Fourier transform in 1.1 is performed on equally spaced points of the complex unit circle, i.e.

$$\omega_k = \frac{2\pi k}{L}, k = 0, 1, \ldots, \frac{L}{2}.$$

L is here the order of the Fourier transform. The reconstructed signal

$$y(n) = \sum_k X_n(e^{j\omega_k})e^{j\omega_k n}$$

is a scaled replica of $x(n)$ according to the relation

$$y(n) = Lw(0)x(n).$$

The so-called Filter-Bank Summation allows perfect reconstruction from STFT coefficients if the order L of the Fourier transform used in 1.1 is larger than the length N of $w(n)$. A necessary condition for the reconstruction to be exact even if $L < N$ is that

$$w(rL) = 0, \forall r.$$

OverLap Addition (OLA)

The main assumption in OLA method is to admit that the analysis window is sampled at a sufficiently dense rate so that holds the relation

$$\sum_m w(m-n) = W(e^{j0}), \forall n.$$

The OLA method builds the signal

$$y(n) = \sum_m \sum_k X_m(e^{j\omega_k})e^{j\omega_k n}$$

that allows the following scaled reconstruction:

$$y(n) = Lx(n)W(e^{j0}).$$

The methods presented above allow time-frequency analysis using short-time Fourier transform, and perfect reconstruction of the temporal signal is made possible with few assumptions on the analysis window. The next paragraph will present more recent time-frequency analysis methods using more algebra. They offer a different interpretation of the STFT coefficients who are considered as scalar products.

1.2.3 Hop size of analysis window

It is possible to define the short-time Fourier transform of a signal $x(n)$ such that the analysis window, of length N, is shifted of H samples. It is necessary that $H \leq N$, that means that two consecutive windows must overlap each other of at least one sample.

$$X_n(e^{j\omega_k}) = \sum_{m=-\infty}^{\infty} w(n - Hm)x(m)e^{-j\omega_k m}.$$

Note that the previous case corresponds to $H = 1$. Two consecutive analysis windows overlap each other of $R = N - H$ samples. This new definition leads to a different sampling rate for the temporal variable of STFT coefficients $(X_n(e^{j\omega_k}))$.

1.3 Theory of Gabor frames

The time-frequency analysis initiated by Gabor has been even more developped later in the 20th century with algebraic tools that allowed different interpretations of what a time-frequency representation *can be* and gave supplementary results in perhaps a more elegant way.

As [Mallat, 2008] and [Kaiblinger, 2005] state, the STFT coefficients (c_{nk}) can be seen as a scalar product instead of a convolution.

$$c_{nk} =< x, g_{nk} >$$

where $g_{nk}(m) = w(n - m)e^{j\omega_k n}$. The scalar product $< \cdot, \cdot >$ is defined as

$$< x, y >= xy^\dagger = \sum_n x(n)y(n)^*$$

for discrete complex $x(n)$ and $y(n)$, the asterisk meaning complex conjugate.

In order to extend such an interpretation of time-frequency analysis, the theory of Gabor frames makes use of a Gabor system, that is defined as a family of functions

$$\begin{cases} G(g, a, b) = \{g_{kl} : k, l \in \mathbb{Z}\} \\ \text{with } g_{kl}(n) = g(n - ka)e^{j2\pi nlb} \end{cases}$$

A Gabor system is a frame if there exist constants $A, B > 0$ such that for any signal x

$$A\|x\|^2 \leq \sum_{k,l \in \mathbb{Z}} | < x, g_{kl} > |^2 \leq B\|x\|^2.$$

One essential condition on a and b for the Gabor family to be a frame is $ab < 1$. Note that in the unified approach of STFT described in latest paragraph, we had $a = H \leq N - 1$ and $b = 1/L$, so that $ab \leq \frac{N-1}{L}$. If $N \leq L$, the STFT is a frame.

Finally, the dual Gabor frame of a Gabor frame g_{kl} is the frame $\tilde{g}_{kl}$ such that

$$x(n) = \sum_{k,l} < x, g_{kl} > \tilde{g}_{kl}(n) = \sum_{k,l} < x, \tilde{g}_{kl} > g_{kl}(n).$$

The dual Gabor frame generalises the inversion of short-term Fourier transform to any type of Gabor frame anaysis. A powerful Matlab toolbox for Gabor analysis called `ltfat` is available online. It has been used for the time-frequency operations in all the numerical simulations presented in this report.

Chapter 2

The phase

2.1 Different phases

The phase is a parameter that allows one to extend the set of real numbers to the set of complex numbers. The phase is related to rotations in the complex plane, thus it links complex numbers or signals to trigonometry and Fourier analysis.

2.1.1 Definition of the phase of a complex number

There are two ways to define the complex numbers which both lie on the unit imaginary number i defined such that

$$i^2 = -1.$$

A complex number z can be written as the sum of its real and imaginary part:

$$z = a + ib, \ (a, b) \in \mathbb{R} \times \mathbb{R},$$

where $a = \Re z$ is the real part of z and $b = \Im z$ is its imaginary part. $\mathbb{C} = \{a + ib, (a, b) \in \mathbb{R} \times \mathbb{R}\}$ is a 2-D real vector space spanned by 1 and i. The real axis $\mathbb{R}$ and the imaginary axis $i\mathbb{R} = \{ib, b \in \mathbb{R}\}$ are vector subspaces of $\mathbb{C}$.

A complex number can also be written as the product of its magnitude and of a phase term.

$$z = \rho e^{i\theta}, \ (\rho, \theta) \in [0, +\infty[\times [0, 2\pi[$$

where $\rho = |z|$ is the magnitude of z and $\theta = \arg z$ is the argument, or phase, of z. The following mapping allows one to switch from the real and imaginary part to the magnitude and phase.

$$\begin{cases} \rho = \sqrt{a^2 + b^2} \\ \theta = \arctan \dfrac{b}{a} \end{cases} \tag{2.1}$$

The second equality in system 2.1 shows that if z is real we have $b = 0$ and $\theta \in \{0, \pi\}$.

2.1.2 Phase of a complex signal

Let s be a signal in a set $\{\mathcal{X} \to \mathbb{C}\}$ where $\mathcal{X}$ can be $\mathbb{Z}$ for discrete signals or $\mathbb{R}$ for continuous ones. The phase of s is the signal $\theta \in \{\mathcal{X} \to [0, 2\pi[\}$ such that

$$\forall x \in \mathcal{X}, \theta(x) = \arg[s(x)]$$

Note that if we consider the set of real signals $\{\mathcal{X} \to \mathbb{R}\}$ all signals will have a zero phase. This is the case for signals such as audio signals that represent a variation of a localized pression field or a variation of voltage. The definition of phase given above becomes interesting when we consider signals with non-zero imaginary part. Complex signals appear when we apply complex operators on real signals, such as the Fourier transform.

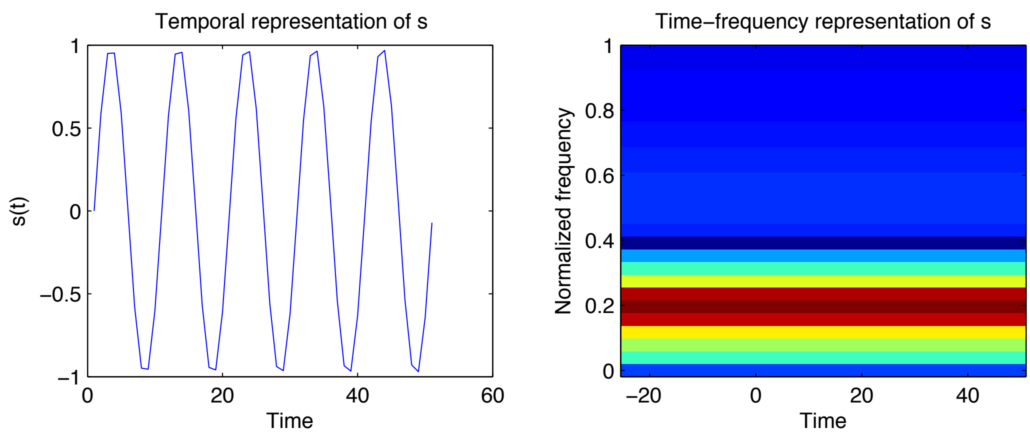

Figure 2.1: Visualization of the spectral components of a sound on the magnitude of STFT

2.1.3 Phase of the Fourier transform of a signal

Continuous signals The Fourier transform of a real signal $s \in L^2(\mathbb{R})$ is the complex signal $\mathcal{F}s \in L^2(\mathbb{R})$ where $\mathcal{F}$ is the Fourier operator such that

$$\mathcal{F}\{s\}(\omega) = \int_{\mathbb{R}} s(t)e^{-2i\pi\omega t}dt, \omega \in \mathbb{R}.$$

It may be separated in its real and imaginary part

$$\mathcal{F}\{s\}(\omega) = \int_{\mathbb{R}} s(t)\cos(2\pi\omega t)dt - i\int_{\mathbb{R}} s(t)\sin(2\pi\omega t)dt.$$

So, the phase of the Fourier transform of s is the signal

$$\theta(\omega) = -\arctan\frac{\int_{\mathbb{R}} s(t)\sin(2\pi\omega t)dt}{\int_{\mathbb{R}} s(t)\cos(2\pi\omega t)dt} \tag{2.2}$$

with the convention $\arctan(\pm\infty) = \pm\frac{\pi}{2}$

Discrete signals For a discrete signal $s \in \mathbb{R}^N$, its Fourier transform is

$$\mathcal{F}\{s\}(k) = \sum_{n=0}^{N-1} s(n)e^{-2i\pi\frac{nk}{N}}, k \in 0 : N-1$$

so that the phase of $\mathcal{F}\{s\}$ is

$$\theta(k) = -\arctan\frac{\sum_{n=0}^{N-1} s(n)\sin(2\pi nk)}{\sum_{n=0}^{N-1} s(n)\cos(2\pi nk)}. \tag{2.3}$$

Relations 2.2 and 2.3 show that the phase θ of the Fourier transform of a real signal s carries information on s, both if this signal is continuous or discrete.

The generalization of the phase to a higher dimension signal set is as straightforward as the generalization of the Fourier transform itself. Most achieved techniques of phase retrieval work on 2-D diffraction patterns.

2.1.4 Phase of the short-term Fourier transform

The short-term Fourier transform, or Gabor transform, can be viewed as a 2-D transform of a 1-D temporal signal which aims to describe its behavior according both to time and frequency. That is why one calls such a signal a time-frequency representation.

For a temporal signal $s(t)$ we will call $\Psi^*\{s\}(\omega, \tau)$ its short-time Fourier transform. Ψ^* denotes the short-time Fourier transform operator, whose conjugate operator is Ψ the inverse short-time Fourier transform operator such that $\Psi\{\Psi^*\{s\}\} = s$. Their expression for continuous signals are given in appendix.

By displaying the magnitude of $|\Psi^*\{s\}(\tau, \omega)|$ with a colour code, one can see the variations of the spectral composition of a signal. An example is given on figure 2.1. The analyzed signal is $s = \sin\left(2\pi \frac{f}{F_e}(0:50)\right)$, where $f = 440$ Hz is a constant frequency and $F_e = 4410$ Hz is the sampling frequency. The time-frequency representation on the right shows the magnitude of $|\Psi^*(\tau, \omega)|$ and was obtained using the Matlab command `specgram`. We can see the time-constant frequency component at the normalized frequency $\frac{2f}{Fe} \approx 0.2$.

The time-frequency representation is a representation of the magnitude of the short-time Fourier transform. It is a convenient way to represent musical signals because it allows one to have both the frequency components of the sound and their variation across time, up to the Heisenberg inequality. As it does not display any phase information, not all the information about the signal is easily accessible. In particular, the notion of analytical phase of a real signal that will be defined in the next paragraph will allow one to build signals with exactly the same magintude of short-time Fourier transform.

2.1.5 Phase of real signals

In this section we consider a continuous real signal s. It is possible to define a signal that could hold for the phase of s by building an imaginary part from the real signal. One operation that allows to switch from a real signal to a complex signal is the Hilbert transform.

The assumption that we make concerning the real signal s is that it is the real part of a complex signal $f(z) \in \mathbb{C}, z \in \mathbb{C}$ that is analytic on the upper plane $\{\Re z \geq 0\}$ and that we take along the real axis. The assumption that f is analytic leads, through the Cauchy-Riemann equations, to a strong relation between the real and imaginary parts of f: the imaginary part of f is the Hilbert transform of the real part. Call $\mathcal{H}$ the real linear operator that couples a real part of such a function to its imaginary part. It is defined as the Cauchy principal value of the convolution with the $\left(\frac{1}{\pi t}\right)$ distribution.

$$\mathcal{H}\{s\}(t) = p.v. \left(\frac{1}{\pi t}\right) * s(t) = p.v. \frac{1}{\pi} \int_{\mathbb{R}} \frac{s(t')}{t - t'} dt'$$

The following relation between s and f restricted to the real axis holds.

$$f = s + i\mathcal{H}\{s\} = (Id + i\mathcal{H})\{s\}$$

The complex signal f is called the analytic signal associated to s.

Once this operation defined, it becomes possible to define the phase θ of a real signal by taking the phase of its associated analytic signal.

$$\theta = \arg(s + i\mathcal{H}\{s\}) = \arctan \frac{\mathcal{H}\{s\}}{s}$$

The most important example of analytical signal is also very often used in domains such that electronics, electromagnetics and acoustics. One can show that the Hilbert transform of the cosine function is the sine function, so that the real function

$$s(t) = \cos(t), t \in \mathbb{R}$$

has for associated analytical signal the function

$$f(z) = e^{iz}, z \in \mathbb{C}$$

which is analytical on the whole complex plane.

For the given real signal

$$s(t) = \cos(2\pi ft + \phi), t \in \mathbb{R}$$

where $f \in \mathbb{R}$ is a constant frequency and $\phi \in [0, 2\pi[$ is also a constant, the phase θ is given by

$$\theta(t) = \arctan \frac{\mathcal{H}\{s\}(t)}{s(t)} = \arctan \frac{\sin(2\pi ft + \phi)}{\cos(2\pi ft + \phi)} = 2\pi ft + \phi$$

It allows one to properly define the phase at the origin as

$$\theta(0) = \phi$$

It is straightforward to build the analytic signal associated to any periodic signal using a Fourier series decomposition.

2.2 Relation between the phases

It may be interesting to show off some relations linking the previous definitions of the phases to figure out that phase and amplitude are correlated through the different representations of a signal: temporal, Fourier, or time-frequency.

Theorem 2.2.1. *Let $s \in L^2(\mathbb{R})$ be a complex analytical signal*

$$s(t) = x(t) + i\hat{x}(t) = \rho(t)e^{i\theta(t)}.$$

Let the Fourier transform of x be

$$F(\omega) = R(\omega)e^{i\alpha(\omega)}.$$

The following relations between the phases $\theta(t)$ and $\alpha(\omega)$ hold.

$$\alpha(\omega) = \arctan \frac{\int_{\mathbb{R}} \rho(t)\sin(\theta(t) - 2\pi\omega t)dt}{\int_{\mathbb{R}} \rho(t)\cos(\theta(t) - 2\pi\omega t)dt}$$

$$\theta(t) = \arctan \frac{\int_{\mathbb{R}} R(\omega)\sin(\alpha(\omega) + 2\pi\omega t)d\omega}{\int_{\mathbb{R}} R(\omega)\cos(\alpha(\omega) + 2\pi\omega t)d\omega}$$

Let the short-term Fourier transform of s be

$$S_w(\tau, \omega) = W(\tau, \omega)e^{i\phi(\tau,\omega)}$$

where w is the window function. The following relations between the phases $\theta(t)$ and $\phi(\tau, \omega)$ hold.

$$\phi(\tau, \omega) = \arctan \frac{\int_{\mathbb{R}} \rho(t)w(t-\tau)\sin(\theta(t) - 2\pi\omega t)dt}{\int_{\mathbb{R}} \rho(t)w(t-\tau)\cos(\theta(t) - 2\pi\omega t)dt}$$

$$\theta(t) = \arctan \frac{\int_{\mathbb{R}}\int_{\mathbb{R}} W(\tau,\omega)w(t-\tau)\sin(\phi(\tau,\omega) + 2\pi\omega t)d\omega d\tau}{\int_{\mathbb{R}}\int_{\mathbb{R}} W(\tau,\omega)w(t-\tau)\cos(\phi(\tau,\omega) + 2\pi\omega t)d\omega d\tau}$$

Therefore, we can give similar relations for discrete signals. We then have the relations between the phase of a real signal - of its associated analytic signal - and the phase of its Fourier transform and of its short-term Fourier transform.

2.3 Phase-shifted signals

Definition 2.3.1 (Phase-shifted signal)**.** Let $x \in L^2(\mathbb{R})$ be a real signal that admits a Hilbert transform $\hat{x}$ and $\phi \in \mathbb{R}$. We define the phase-shifted signal x_ϕ as

$$x_\phi = \Re[(x + i\hat{x})e^{i\phi}]$$

The following properties immediately hold.

1. $x_\phi(t) = x(t)\cos\phi - \hat{x}(t)\sin\phi$

2. $\hat{x}_\phi(t) = \hat{x}(t)\cos\phi + x(t)\sin\phi$

3. $\hat{x}_\phi = x_{\phi-\frac{\pi}{2}}$

A description of the Hilbert transform and of its properties is given in appendix C. Most results hold for continuous signals. The discrete case remains a problem discussed among others in [Gold et al., 1969] and [Zhechev, 2005].

*

Chapter 3

The Griffin-Lim algorithm

3.1 Framework of Griffin-Lim algorithm

Suppose we have a magnitude spectrogram $W(\tau, \omega) = |Y(\tau, \omega)|$ but we do not know the phase $\arg Y(\tau, \omega)$. The objective of a phase retrieval algorithm is of course to retrieve a temporal signal whose STFT is as close as possible to $Y(\tau, \omega)$ according to a distance measure. Over the iterations, the distance measure has to become non-increasing. The Griffin-Lim algorithm is a phase-retrieval algorithm presented in [Griffin and Lim, 1984]. The framework of the algorithm is presented on figure 3.1. The input of the algorithm is a time-frequency representation initialized at $S_0 = W$. The algorithm then consists in two projections in the time-frequency domain.

1. The first projection consists of an inverse STFT followed by a STFT.

$$S'_k = \Psi^* \Psi S_k$$

   It is a projector because of the relation

$$\Psi \Psi^* = Id$$

   that holds if the short-time Fourier transform and its inverse are correctly implemented, essentially if the analysis and synthesis windows allow proper reconstruction.

2. The second projection consists in keeping the phase of the spectrogram obtained by the first projection and replace its magnitude by W.

$$S_{k+1} = W e^{i \arg S'_k}$$

So, an iteration of the Griffin-Lim algorithm framework can be summed up in the following formula:

$$S_{k+1} = W e^{i \arg \Psi^* \Psi S_k}$$

It is shown that the L^2-distance between $|S'_k|$ and W is non-increasing:

$$\| |S'_{k+1}| - W \|_2^2 \leq \| |S'_k| - W \|_2^2,$$

so that the algorithm converges to a solution. Note that this distance measure is defined on magnitude data only. It does not take into account the accuracy of phase reconstruction, because we assume that the original phase is unknown.

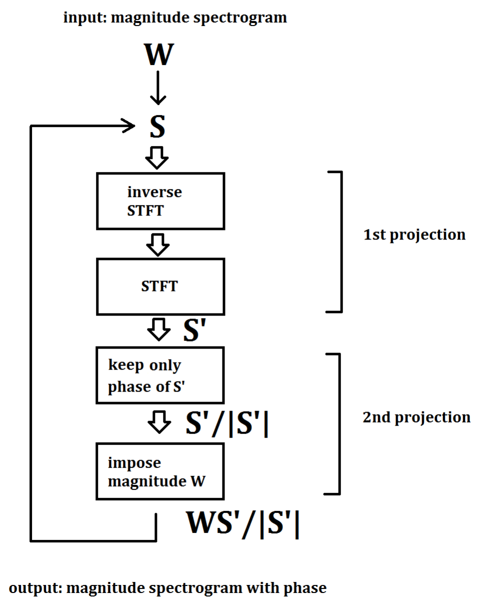

Figure 3.1: Framework of the Griffin-Lim phase retrieval scheme

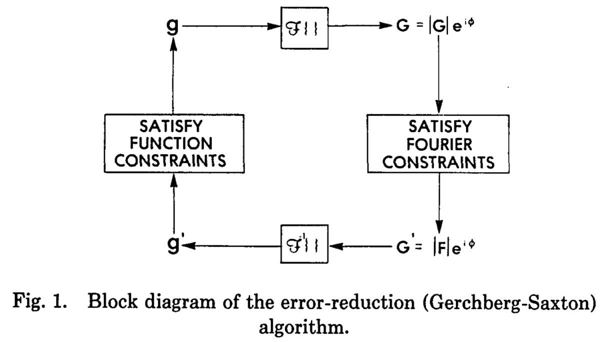

Figure 3.2: Framework of Gerchberg-Saxton algorithm by Fienup [Fienup, 1982]

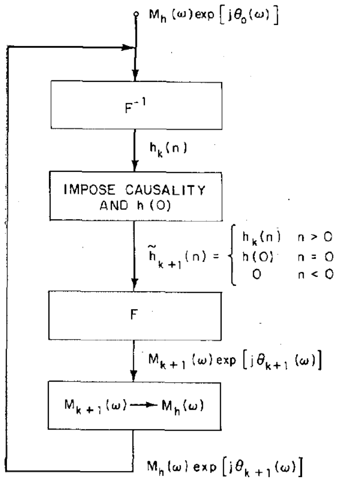

Fig. 2. Iterative algorithm to recover $h(n)$ from its magnitude.

Figure 3.3: Minimum phase iterative algorithm to recover a signal from its magnitude [Quatieri and Oppenheim, 1981]

3.2 Other phase retrieval techniques

3.2.1 The Gerchberg-Saxton algorithm

The Griffin-Lim algorithm is similar to other phase retrieval techniques developped in the 70s and the 80s. The Gerchberg and Saxton's 1971 fundamental article on phase retrieval presents "A practical algorithm for the determination of phase from image and diffraction plane pictures" [Gerchberg and Saxton, 1971]. The framework of the algorithme is breifly summed up in figure 3.2. It allows a reconstruction if one has intensity measurements in both the spatial and Fourier domain [Fienup, 1982]. That is why this algorithm was designed to work on image measurements like diffraction patterns for cristallography. The first phase retrieval techniques were thus applied on intensity measurements of 2-D signals.

3.2.2 The Quatieri-Oppenheim minimum phase algorithm

Before Griffin and Lim, Thomas Quatieri and Alan Oppenheim proposed in 1980 iterative techniques for minimum phase signal reconstruction form phase or magnitude. The unicity of the reconstruction is due to a minimum phase condition performed by imposing causality on the temporal reconstructed signal. The framework of the algorithm they proposed in [Quatieri and Oppenheim, 1981] is described on figure 3.3.

The three methods presented above – Griffin-Lim, Gerchberg-Saxton and Quatieri-Oppenheim – are based on alterning projections, which has been the mainstream method for phase retrieval since early 70s. In [Bouvrie and Ezzat, 2006], the MIT researchers Jake Bouvrie and Tony Ezzatt proposed an incremental algorithm for signal reconstruction from signal reconstruction from short-term Fourier transform magnitude based on numerical root-finding combined with explicit smoothness assumptions.

3.3 Non convexity of Griffin-Lim algorithm

3.3.1 Example on a monochromatic signal

Despite the fact that the Griffin-Lim cost function is decreasing and thus ensuring that the algorithm always converges towards a solution, this solution is not unique because the cost function is not convex. For a given STFT magnitude, the algorithm retrieves a phase of STFT among different phases that it could have retrieved under different initial conditions. To better understand this assumption, consider a discrete sinusoidal signal $x(n)$ with an arbitrary phase at the origin ϕ_x :

$$x(n) = \exp\left(2j\pi f \frac{n}{Fe} + j\phi_x\right), n \in \{0, \cdots, -1\}.$$

In the simulations we shall work on a frequency of $f = 1000$ Hz. Define the phase at the origin of $x(n)$ as

$$\phi_x = \arg(x(0)).$$

Assume we only know the STFT magnitude of $x(n)$ and that we retrieve a temporal signal $y(n)$ from it using the Griffin-Lim algorithm. The phase at the origin of $y(n)$ is

$$\phi_y = \arg(y(0)).$$

The non-convexity of Griffin-Lim leads to different phases at the origin for the original and the reconstructed signals: $\phi_y \neq \phi_x$. We are going to calculate several signal-to-noise ratios in order to measure the error of reconstruction. They can be calculated both on temporal or spectral signals. The error of reconstruction stands for the noise. The signal-to-noise ratio between original and reconstructed temporal signals is also small:

$$R_1 = 10 \log\left(\frac{\|x\|_2^2}{\|x - y\|_2^2}\right) \ll \infty$$

A value of ϕ_y too different than ϕ_x may lead to a negative value of R_1, although both signals have the same STFT magnitude. A Matlab simulation with a randomly picked $\phi_x = 0.4586$ can give a retrieved phase at the origin $\phi_y = -0.9006$. The signal-to-noise ratio is thus $R_1 = -1.8907$ dB.

We expect that the reconstructed signal is a complex exponential with a phase at the origin ϕ_y instead of ϕ_x. Under this assumption, the signal

$$z(n) = \exp\left(j\phi_x - j\phi_y\right) y(n)$$

should be closer to $x(n)$ than $y(n)$, so that

$$R_2 = 10 \log\left(\frac{\|x\|_2^2}{\|x - z\|_2^2}\right) > R_1.$$

On figure 3.4 are plotted the real parts of x, y and z for $\phi_x = 0.458$. The signal z is much closer to x than y and the corresponding signal-to-noise ratio is

$$R_2 = 28.6635 \text{ dB} > R_1 = -1.8907 \text{ dB}$$

Therefore, we find $\phi_z = \phi_x$.

This example on a complex exponential allows one to understand an inconvenient of the Griffin-Lim algorithm. The complex exponentials are built up to a phase shift that we do not know when we lose the phase information of the STFT, because the phase of a complex temporal signal and the phase of its STFT magnitude (or of its Fourier magnitude) are of course related to each other. This example applies for a signal with a single frequency component but could be extended to any kind of signal, assuming that we find the accurate transformation to properly shift the phase and enhance the reconstruction. The code used to do the simulation is `non_convexity_of_griffin_and_lim.m`, given in the appendix of this chapter.

3.3.2 Sensitivity to initial conditions

For a complex monochromatic signal

In this section we compute the code `non_convexity_of_griffin_and_lim_2.m` to make a loop on 50 values of the origin phase ϕ_x linearly varying between 0 and 2π. We get the corresponding value of the retrieved phase at the origin and we want to see how a change of phase at the origin of the original signal affects the retrieved signal. It is possible to plot both ϕ_x and ϕ_y varying across the iterations of the loop, as shown on figure 3.5, or to plot ϕ_y as a function of ϕ_x as shown on figure

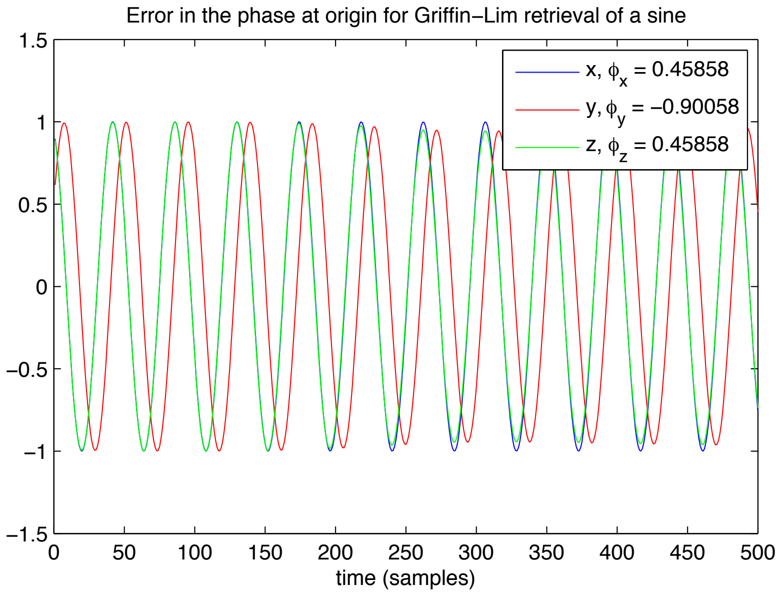

Figure 3.4: Reconstruction y of a complex exponential x by Griffin-Lim algorithm and corrected signal z. Real parts are plotted.

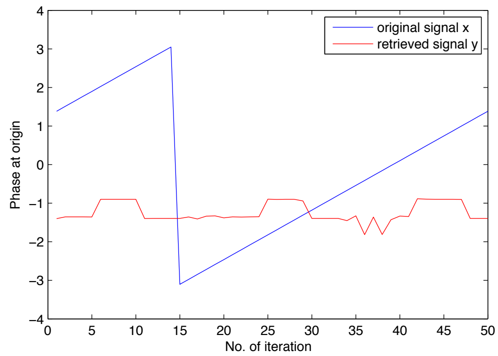

Figure 3.5: Phase at the origin ϕ_x of a complex exponential signal and retrieved phase ϕ_y using Griffin-Lim algorithm

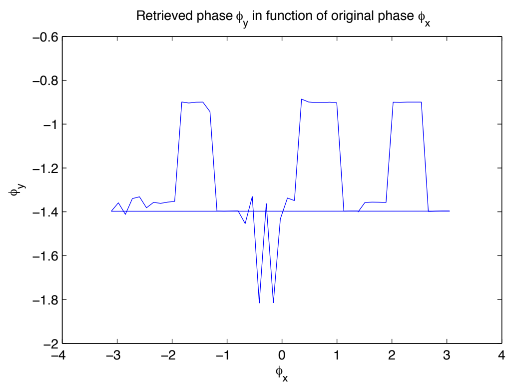

Figure 3.6: Griffin-Lim retrieved phase at the origin ϕ_y varying across original phase ϕ_x for a complex exponential monochromatic signal

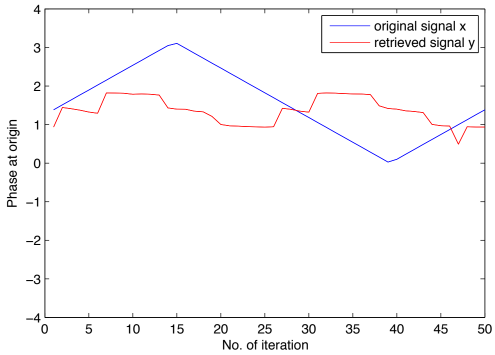

Figure 3.7: Phase at the origin ϕ_x of a real monochromatic signal and retrieved phase ϕ_y using Griffin-Lim algorithm

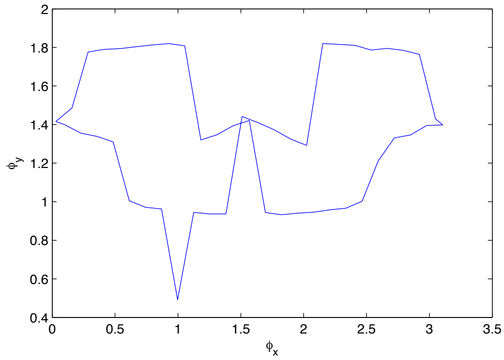

Figure 3.8: Griffin-Lim retrieved phase at the origin ϕ_y varying across original phase ϕ_x for a real monochromatic signal

3.6. The two figures show that ϕ_y does not linearly depend on ϕ_x : the values of ϕ_y are bounded in a slower interval, approximately [-2, 0.5] while the original phase ϕ_x fluctuates from $-\pi$ to π. It seems obvious that different values of the same phase at the origin can give the same retrieved phase at the origin, which leads in the case of a pure sine to the same signal.

For a real monochromatic signal

If we do the simulations only on the real part of the previous signal, because the final issue is to retrieve real audio signals, it is possible to compute the phase at the origin as

$$\begin{cases} \phi_x = \arccos(x(0)) \\ \phi_y = \arccos(y(0)). \end{cases}$$

The results of the simulations are presented on figures 3.7 and 3.8. The results are different from the complex case. One can think that the complex case allows less calculation errors as the inverse STFT is not truncated to its real values. However, in both simulations it seems that there is no correlation between the retrieved phase at the origin and the original one.

This simulation was done on a monochromatic signal where the phase is an easy concept to define using basic trigonometry. On an actual sound signal the notion of phase is harder to define. One can use the Hilbert transform but its implementation on discrete, finite signals is not very accurate and involves boundary problems that are the cause of a poor precision.

Chapter 4

Convex optimization for phase retrieval

4.1 Convex optimization

4.1.1 Quick overview of convex optimization history

The convex optimization methods arose from fundamental papers written in the 1960s and in the 1970s. Amongst them, the French mathematician Jean-Jacques Moreau studied the notions of distance in a Hilbert space[Moreau, 1965]. He introduced in 1963 the proximal operator [Capricelli, 2008, p.12].

In the 1970s the American mathematician R. Tyrell Rockafellar brought major contribution to convex optimization methods. Almost at the same time, the French mathematician Jean-Baptiste Hiriart-Urruty published papers on optimality conditions in nondifferentiable programming [Hiriart-Urruty, 1978].

The convex optimization theory was summed up by Boyd and Vandenberghe who gave courses at Stanford university and published in 2004 a book available online for free [Boyd and Vandenberghe, 2004].

Convex optimization ensures that the algorithm will converge towards a unique value if the constraint functions and the considered vector set for convergence are convex. Note that the constraint functions do not need to be differentiable. One method consists in alterning the following steps:

- projecting the vector onto the convex set using a proximal operator,
- projecting the vector onto the constraint set using the constraint function.

This leads to a variety of techniques that is summed up in [Combettes and Pesquet, 2010]. According to this description, every convex optimization algorithm can be seen as an alterning projection algorithm.

4.1.2 The proximal operator

The proximal operator is was introduced by Jean-Jacques Moreau in 1965 in [Moreau, 1965].

Let H be a real Hilbert space and C a non-empty convex subset of H. Let $f : H \to]-\infty, +\infty]$ be a convex and lower semicontinuous and not equal to $+\infty$ everywhere. The proximal operator of f is defined as

$$\operatorname{prox}_f(z) = \arg\min_u \left(\frac{1}{2}\|u - z\|^2 + f(u) \right)$$

4.1.3 Projected gradient descent method

The projected gradient method is presented in [Recht, 2012]. Its goal is to find the minimum of a function $h : H \to]-\infty, +\infty]$ that can be written

$$h(x) = f(x) + P(x)$$

where f is differentiable, ∇f is L-Lipschitz [1] and P is convex. The gradient descent ensures that for all $\nu \in \mathbb{R}$,

$$x_* = \text{prox}_{\nu P}(x_* - \nu \nabla f(x_*))$$

if and only x_* is an argmin of $f(x) + P(x)$. The sequence

$$x_{k+1} = \text{prox}_{\nu P}(x_k - \nu \nabla f(x_k))$$

is an algorithm converging to an argmin of $h(x)$.

4.1.4 Semi-definite programming

Semi-definite programming is a way to relax a combinatorial problem [Mitchell, 2000], i.e. to enlarge the set of solution to reduce the complexity of the algorithms solving the problem. Semi-definite programming solves [Waldspurger et al., 2012] :

$$\begin{array}{ll} \text{mininmize} & \mathbf{tr}(UM) \\ \text{subject to} & diag(U) = 1 \\ & U \succeq 0 \end{array}$$

in the variable $U \in H_n$, where H_n indicates the cone of Hermitian matrices of dimension n. There may be supplementary constraints on U. Semi-definite programs are solved by projected gradient descent methods.

4.2 Convex phase retrieval algorithms

4.2.1 PhaseLift, PhaseCut

PhaseCut [Waldspurger et al., 2012] and PhaseLift [Candès et al., 2011] are two different formulations of convex optimization methods for phase retrieval, i.e. retrieve $x \in \mathbb{C}^n$ such that $|Ax| = b$, $b \in \mathbb{R}^m$.

PhaseCut is formulated as

$$\begin{array}{ll} \text{mininmize} & \mathbf{tr}(BU) \\ \text{subject to} & \mathbf{tr}(MU) = 0 \\ & diag(U) = 1 \\ & U \succeq 0 \end{array}$$

where $B = diag(b)(A^\dagger)^* A^\dagger diag(b)$ and $M = diag(b)(AA^\dagger - I)diag(b)$.

PhaseLift is formulated as

$$\begin{array}{ll} \text{mininmize} & \mathbf{tr}(X) \\ \text{subject to} & \mathbf{tr}(AXA^*) = diag(b)^2 \\ & X \succeq 0 \end{array}$$

Voroninski showed in [Voroninski, 2012] that PhaseLift and PhaseCut are simultaneously exact in the $m \geq n$ regime and that PhaseCut fails at sparse recovery when $m < n$.

PhaseCut is the convex relaxation of the NP-hard MaxCut problem

$$\begin{array}{ll} \text{mininmize} & \mathbf{tr}(BU) \\ \text{subject to} & \mathbf{tr}(MU) = 0 \\ & diag(U) = 1 \\ & U \succeq 0 \\ & \mathbf{rank}(U) = 1, \end{array}$$

the constraint $\mathbf{rank}(U) = 1$ being non-convex.

[1] A function g is L-Lipschitz if for all x and y in the set of definition of g, $\|g(x) - g(y)\| \leq L\|x - y\|$.

4.2.2 STLiFT

In [Sun and Smith, 2012], Sun and Smith adapted PhaseLift to the case where A is a time-frequency operator such as short-time Fourier transform. The problem is still to find a signal x whose STFT has the same magnitude as a given STFT magnitude $|Y|^2$.

$$\begin{array}{ll} \text{find} & x \\ \text{subject to} & |\Psi^*\{x\}(mR,\omega_k)|^2 = |Y(mR,\omega_k)|^2 \end{array}$$

but this time the problem is formulated as depending on the variable $U = xx^T$. They called their algorithm STLiFT.

$$\begin{array}{ll} \text{mininmize} & \mathbf{tr}(U) \\ \text{subject to} & \mathbf{tr}(S_{k,m}U) = |Y(mR,\omega_k)|^2 \\ & 0 \le k < N,\ 0 \le m < M \\ & U \succeq 0 \end{array}$$

They adapt the matrix formulation of PhaseLift to time-frequency notations. This change is actually nothing more than a change of index. $S_{k,m}$ is a $L \times L$ matrix, where L is the length of x. It is defined as

$$S_{k,m} = (W_{mR}^T s_k)(W_{mR}^T s_k)^*$$

where s_k is a Fourier vector, such that

$$s_k(n) = e^{j2\pi nk/N}, 0 \le k < N-1$$

and W_{mR}^T is a $N \times L$ matrix containing $diag(w)$ on columns $(m-1)R+1$ to $(m-1)R+N$.

$$W_{mR} = \begin{pmatrix} 0 & \dots & 0 & w(1) & 0 & \dots & \dots & \dots & 0 \\ \vdots & & & \ddots & \ddots & \ddots & & & \vdots \\ \underbrace{0 & \dots & \dots}_{\text{(m-1)R}} & \underbrace{\dots & 0 & w(n)}_{\text{N}} & 0 & \dots & 0 \end{pmatrix}.$$

The result of such an implementation is that

$$\mathbf{tr}(S_{k,m}xx^T) = |X(mR,\omega_k)|^2.$$

This formulation shows that the norm linearly depends on xx^T. More over, the trace is a convex operator. The distance measure defined by Griffin and Lim as

$$D(x) = \sum_{m,k} \left(|X(mR,\omega_k)|^2 - |Y(mR,\omega_k)|^2\right)^2$$

is expressed as follows by Sun and Smith:

$$D(xx^T) = \sum_{m,k} \left(\mathbf{tr}(S_{k,m}xx^T) - |Y(mR,\omega_k)|^2\right)^2.$$

Projected gradient descent method The minimization of the distance measure is computed using a projected gradient descent method. Sun and Smith define the Lagrangian associated to the distance measure:

$$\mathcal{L}(U) = D(U) + \lambda\mathbf{tr}(U)$$

where λ is a parameter that is supposed to decrease to zero. It is called a Lagrange multiplier. It "controls the tradeoff between minimizing the distance between the spectrograms and the low-rank constraint" on U [Sun and Smith, 2012]. The use of Lagrange multipliers fastens the convergence but does not change the limit. In order to take into account the constraint on U to be semi-definite positive, it is possible to write the function that we are to minimize as

$$h(U) = \mathcal{L}(U) + ind(U)$$

where ind is the indicator function of the cone of semi-definitive positive matrices, defined as

$$ind(U) = \begin{cases} 0 \text{ if } U \text{ is SDP} \\ +\infty \text{ if } U \text{ is not SDP.} \end{cases}$$

$L(U)$ is convex, differentiable and assume its gradient is C-Lipschitzian. The proximal operator of the indicator function of a subset is the projection onto this subset. So the iterative scheme for convergence is

$$U_{k+1} = \mathcal{P}\left(U_k - \frac{1}{C}\nabla\mathcal{L}(U_k)\right)$$

where $\mathcal{P}$ is the projector onto the set of semi-definite positive matrices.

Initial value U_0 The program being convex, it can start from any initial value U_0. For instance, it can be $U_0 = 0$ or $U_0 = x_0 x_0^T$ with $x_0 = \Psi\{|Y|\}$.

Gradient of the Lagrangian The gradient of the Lagragian is

$$\nabla\mathcal{L}(U) = 2\sum_{k,m}\left(\mathbf{tr}(S_{k,m}U - |Y(mR,\omega_k)|^2\right) S_{k,m} + \lambda I.$$

Lipschitz constant of the gradient The Lipschitz constant of the gradient is computed using a power iteration method.

Let Z_0 be a real random matrix of size $L \times L$. Then the iteration scheme

$$Z_{k+1} = \frac{\nabla\mathcal{L}(Z_k)}{\|Z_k\|}$$

makes the sequence $\|Z_k\|$ converge to the greatest eigenvalue of the matrix operator $\nabla\mathcal{L}$ that is a majorant the Lipschitz constant of $\nabla\mathcal{L}$. As we are likely to solve the problem for the Fröbenius norm, we use this one in the computation of C.

Projection $\mathcal{P}$ onto the set of positive semi-definitive matrices This is the delicate part of the algorithm, because it needs find the eigenvalues of a $L \times L$ matrix, and keep only the positive values. If U can be written

$$U = M diag(\lambda_1, \ldots, \lambda_L)M^{-1}$$

then $\mathcal{P}(U)$ is

$$\mathcal{P}(U) = M diag(\max(0,\lambda_1), \ldots, \max(0,\lambda_L))M^{-1}.$$

This operation makes the convex optimization method very slow and currently not usable on actual size audio signals.

Lagrange multipliers As explained above, the Lagrange multipliers are to decrease to zero so that the relaxation is efficient. For the numerical simulations, we have made them decrease on a logarithmic scale. Here is an extract of the Matlab code used for the simulations explaining how we made use of the Lagrange mutlipliers :

```
ss = real(ISTFFT(abs(X),w,R));
    for lambda = logspace(0,-4,20);
        t = 1/lip_const_grad(P,w,H,lambda);
        ss = stlift(P,w,H,lambda,t,ss,eps);
        ss = ss*norm(x)/norm(ss);
    end
```

`lip_const_grad` computes the Lipschitz constant of the gradient, `stlift` performs the projected gradient descent method with Lagrange multiplier `lambda` [2]

[2]The author wishes to thank Dennis Sun for his help throughout the ocean. For the simulations, the author used the code that Dennis Sun sent him because it was much faster than the one he painfully wrote himself.

Chapter 5

Comparison of Griffin-Lim and Sun-Smith algorithms

5.1 Criteria for comparison

In this chapter we make several Matlab simulations to compare the two considered algorithms of phase retrieval from STFT magnitude: the alterning projections algorithm of Griffin-Lim and the newer convex optimization method of Sun-Smith. We expect Griffin-Lim to be non convex, so not very precise, and Sun-Smith to need much computation time. For this last reason, we are going to work on short signals of $L = 16$ and $L = 32$ samples. We are testing two types of windows : Hann and Gauss, different lengths of windows N and different overlaps from $0.1N$ to $0.9N$.

We denote by x the original temporal signal that we take real, X its STFT, y the temporal reconstructed signal and Y its STFT. We are interested in several parameters that we will compute both for Griffin-Lim and Sun-Smith.

1. The temporal signal-to-noise ratio between the original and reconstructed signals

$$R_1 = 10 \log \left(\frac{\|x\|_2^2}{\|x - y\|_2^2} \right).$$

2. We want to compare the signal-to-noise ratio R_1 to the following modified temporal signal-to-noise ratio:

$$R_2 = 10 \log \left(\frac{\|x\|_2^2}{\|min(|x - y|, |x + y|)\|_2^2} \right).$$

   A reconstruction with a good R_1 will have a good R_2. A reconstruction with a poor R_1 but with a good R_2 means that at least part of the reconstructed signal is the opposite of the corresponding part of the original signal.

$$R_2 > R_1 \Rightarrow \exists n \in \{0, \cdots, L-1\}, y(n) \approx -x(n)$$

   Of course, a reconstruction with both poor R_1 and R_2 is just a bad reconstruction.

3. The spectral signal-to-noise ratio

$$R_3 = 10 \log \left(\frac{\|X\|_2^2}{\|X - Y\|_2^2} \right).$$

   Note that if the reconstruction is tight, it preserves the norm, so we should have $R_1 = R_3$.

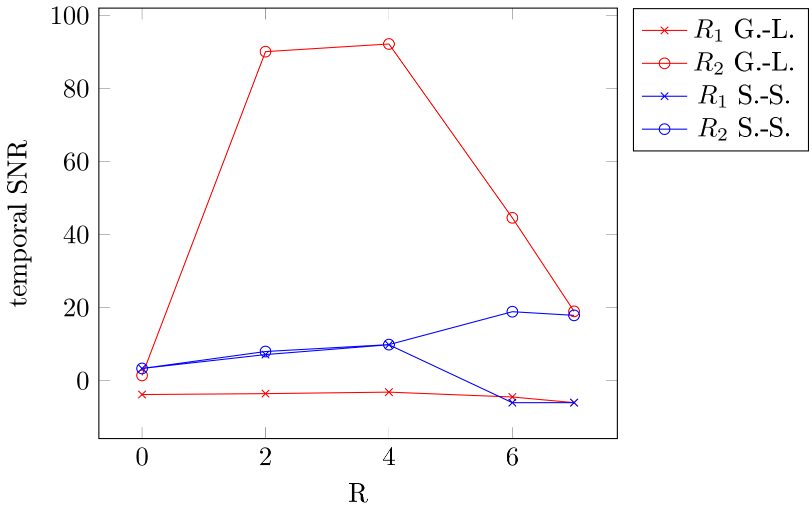

Figure 5.1: Example of temporal SNR for reconstruction of a random signal of $L = 16$ samples with a window of length $N = 8$ for different overlaps R.

4. The spectral signal-to-noise ratio of magnitudes R_4

$$R_4 = 10 \log \left(\frac{\|X\|_2^2}{\||X| - |Y|\|_2^2} \right).$$

Note that as the distance measure is built with the magnitudes of STFT and is actually decreasnig, we expect a very good convergence of $|X|$ to $|Y|$, so very high values for R_4 in both Griffin-Lim and sun-Smith cases.

5. The Itakura-Saito divergence

$$D_{IS} = \frac{1}{2\pi} \sum_{n,\omega} \left(\frac{|X(n,\omega)|}{|Y(n,\omega)|} - \log \frac{|X(n,\omega)|}{|Y(n,\omega)|} - 1 \right)$$

6. The speed of convergence v defined as the ratio between the number L of samples of the temporal signal and the Matlab calculation time t given by `tic-toc` :

$$v = \frac{L}{t}.$$

All the results of the simulations are presented in appendix A. In the section below we present a particular case that allows one to understand what are the main differences between the two algorithms and then how to read the other data.

5.2 Comparison of the algorithm on a random example

On figure 5.1 the temporal SNR R_1 and R_2 are shown for different overlaps at fixed length of analysis. For the same signal, the spectral SNR R_3 and R_4 are given on figure 5.2. Griffin-Lim performs a better reconstruction of the signal up to sign errors R_2 but Sun-Smith is more precise for the actual mean-square distance R_1.

For spectral SNR this difference is even more empahsized, because the SNR of STFT magnitudes R_4 is around 300 for Griffin-Lim. This is the same order of error that we get when calculating the SNR between x and $\Psi\Psi^* x$, where $\Psi\Psi^* \approx Id$ up to numerical noise. Note that for $R = 0$, Griffin-Lim gave $R_4 = \infty$. That is why such a value does not appear on the graph.

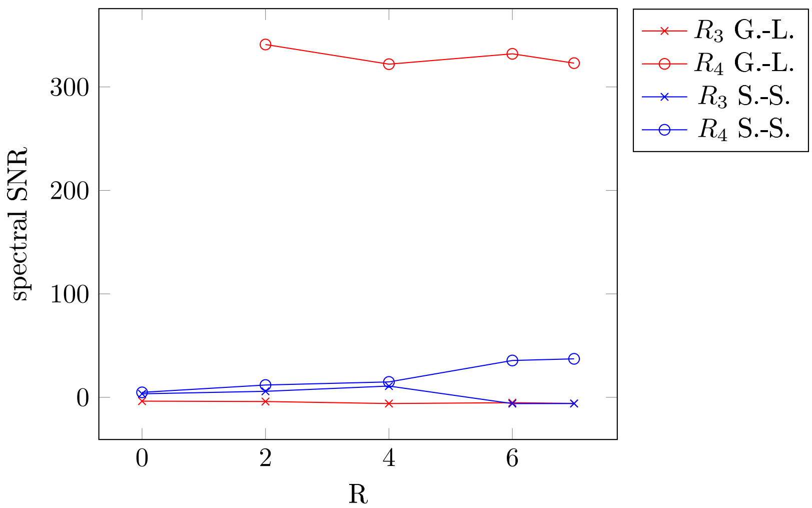

Figure 5.2: Example of spectral SNR for reconstruction of the same random signal of $L = 16$ samples with a window of length $N = 8$ for different overlaps R.

Sun-Smith algorithm does not reconstruct the magnitudes as well as Griffin-Lim, but its value of R_3 that takes the phase into account is slightly higher than the Griffin-Lim value.

5.3 Further remarks concerning the comparison tables

About the Itakura-Saito divergence The use is made of the Itakura-Saito distance that is a percpetual difference between spectra. It only takes into account the magnitudes of the original and reconstructed spectra. We note that it is often zero with Griffin-Lim reconstruction because the convergence in magnitude is better than what we observed for Sun-Smith.

About the size of the overlap The tables of comparison show that for Griffin-Lim, the reconstruction gets better as the overlap increases. Indeed, as explained in [Sturmel and Daudet, 2011], a big overlap avoids stagnation. The size of overlap has not such an influence of the convex reconstruction.

About the speed of calculation The speed of calculation was defined below as the ratio between the length of the signal and the elapsed time given by Matlab between the beginning and the end of the computation of each method. It is surprizing that the Griffin-Lim algorithm does not seem much faster than Sun-Smith, according to those results. For some results it was even slower. Remind that the simulations were made on very small signals ($L = 16$ and 32 samples only) and it took a long time to perform them: around one hour and a half for every four tables. It seems to the author that Sun-Smith calculations took most of this time, but such a result does not clearly appear in the numerical results, maybe due to a bad estimation of the speed of calculation. Moreover, the Sun-Smith algorithm used comutational optimization to be faster, while the Sun-Smith algorithm did not. The personal conclusion of the author is that the Sun-Smith algorithm is much slower than Griffin-Lim, although the results of the simulation do not show it radically.

Chapter 6

Study of the phase retrieval in the Grffin-Lim algorithm

In this chapter we study how the Griffin-Lim algorithm builds a monochromatic signal from its short-time Fourier transform in order to understand how it retrieves the phase. We being by theoretical calculations on the differences between the STFT squared magnitude of a cosine signal and the STFT squared magnitude of the same signal shifted in phase. This difference is non-zero. We already have seen that the Griffin-Lim algorithm seems to retrieve a cosine signal up to a phase term. Actually, it cannot be a perfect cosine signal due to the very little error between the STFT magnitudes. So, Griffin-Lim algorithm fails in the phase retrieval of a sine.

6.1 Comparison of STFT magnitudes of a sine and of a cosine

In order to understand how the phase retrieval from STFT magnitude of a sine works, we are going to calculate the difference of magnitude between the STFT of a sine and the STFT of a cosine.

Define the discrete signals

$$\begin{cases} s_1(n) = \cos(2\pi f n), n \in (0 : L-1) \\ s_2(n) = \sin(2\pi f n), n \in (0 : L-1) \end{cases}$$

and their short-term Fourier transforms

$$S_1(n,k) = \sum_{m=0}^{N-1} \cos(2\pi f m) w(m - Hn) e^{2i\pi \frac{mk}{N}}$$

$$S_2(n,k) = \sum_{m=0}^{N-1} \sin(2\pi f m) w(m - Hn) e^{2i\pi \frac{mk}{N}}.$$

Separate real and imaginary parts in order to calulate the magnitudes.

$$S_1(n,k) = \sum_{m=0}^{N-1} \cos(2\pi f m) w(m - Hn) \cos\left(2\pi \frac{mk}{N}\right) + i \sum_{m=0}^{N-1} \cos(2\pi f m) w(m - Hn) \sin\left(2\pi \frac{mk}{N}\right),$$

so

$$
\begin{aligned}
|S_1(n,k)|^2 &= \left(\sum_{m=0}^{N-1} \cos(2\pi f m) w(m-Hn) \cos\left(2\pi \frac{mk}{N}\right)\right)^2 + \left(\sum_{m=0}^{N-1} \cos(2\pi f m) w(m-Hn) \sin\left(2\pi \frac{mk}{N}\right)\right)^2 \\
&= \sum_{l,m=0}^{N-1} \cos(2\pi f l) \cos(2\pi f m) w(l-Hn) w(m-Hn) \\
&\quad \times \left(\cos\left(2\pi lk/N\right) \cos\left(2\pi mk/N\right) + \sin\left(2\pi lk/N\right) \sin\left(2\pi mk/N\right)\right) \\
&= \sum_{l,m=0}^{N-1} \cos(2\pi f l) \cos(2\pi f m) w(l-Hn) w(m-Hn) \cos\left(2\pi k \frac{l-m}{N}\right).
\end{aligned}
$$

Similarly,

$$
|S_2(n,k)|^2 = \sum_{l,m=0}^{N-1} \sin(2\pi f l) \sin(2\pi f m) w(l-Hn) w(m-Hn) \cos\left(2\pi k \frac{l-m}{N}\right).
$$

So, the difference between the magnitudes can be written

$$
|S_2(n,k)|^2 - |S_1(n,k)|^2 = \sum_{l,m=0}^{N-1} \cos(2\pi f(l+m)) w(l-Hn) w(m-Hn) \cos\left(2\pi k \frac{l-m}{N}\right). \tag{6.1}
$$

This difference between magnitudes is generally different from zero. It is bounded:

$$
\left| |S_2(n,k)|^2 - |S_1(n,k)|^2 \right| \leq \left(\sum_{m=0}^{N-1} w(m-Hn)\right)^2 \leq \|w\|_2^2. \tag{6.2}
$$

The first inequality becomes an equality when we have simultaneously

$$
\begin{cases} f \in \mathbb{Z}, \\ \frac{k}{N} \in \mathbb{Z}. \end{cases}
$$

Those constraints are on frequency variables, f being the constant frequency of the signal, k being the number of the frequency channel.

6.2 Extension to a any phase shift on a monochromatic signal

It is now straightforward to calculate the difference of magnitude between the STFT of a monochromatic signal and the STFT of the same signal shifted of a phase term ϕ. This difference can be expressed as a function of S_1, S_2 and ϕ.

Let s_ϕ be

$$
\begin{aligned}
s_\phi(n) &= \cos(2\pi f n + \phi) \\
&= s_1(n)\cos(\phi) - s_2(n)\sin(\phi).
\end{aligned}
$$

The STFT begin a linear transform, the STFT of s_ϕ is

$$
S_\phi(n,k) = S_1(n,k)\cos(\phi) - S_2(n,k)\sin(\phi)
$$

and its magnitude can be written

$$
\begin{aligned}
|S_\phi|^2 &= |S_1|^2 \cos^2(\phi) - |S_2|^2 \sin^2(\phi) - 2\Re(S_1 S_2^*) \cos(\phi)\sin(\phi) \\
&= |S_1|^2 + (|S_2|^2 - |S_1|^2)\sin^2(\phi) - 2\Re(S_1 S_2^*) \cos(\phi)\sin(\phi)
\end{aligned}
$$

with $|S_2|^2 - |S_1|^2$ as calculated in equation 6.1, and

$$\Re(S_1(n,k)S_2(n,k)^*) = \sum_{l,m=0}^{N-1} \cos(2\pi f m)\sin(2\pi f l) w(m-Hn) w(l-Hn) \cos\left(2\pi k \frac{m+l}{N}\right).$$

This quantity is also bounded:

$$|\Re(S_1(n,k)S_2(n,k)^*)| \le \left(\sum_{m=0}^{N-1} w(m-Hn)\right)^2 \le \|w\|_2^2$$

so that the difference between $|S_\phi|^2$ and $|S_1|^2$ is bounded depending on ϕ:

$$\left||S_\phi|^2 - |S_1|^2\right| \le \left|\sin^2(\phi) - \sin(2\phi)\right| \left(\sum_{m=0}^{N-1} w(m-Hn)\right)^2 \le \left|\sin^2(\phi) - \sin(2\phi)\right| \|w\|_2^2.$$

In the next paragraph, we will see that it possible to calculate the difference of magnitude between the STFT of any signal that admits a Hilbert transform and the same signal shifted of a phase term ϕ that acts on the phase of the associated analytical signal.

6.3 Characterization of the error caused by a shift phase on the STFT magnitude of a pure sine

6.3.1 Motivation

In this section we give an answer to the question:

What is the consequence of a temporal phase shift on the magnitude of STFT?

The element of answer that we bring may lead in the chpater on Griffin-Lim algorithm to what may appear as a contradiction. The Griffin-Lim algorithm builds temporal signals from their STFT magnitude. Observations show that from the STFT magnitude of a sine, it builds something looking like a sine but with a different phase at the origin. The original and reconstructed temporal signals are very different in a least-square sense, but their STFT magnitudes are very close one from each other. The reconstructed temporal signal with the phase at the origin of the original is much closer to the original. We thus wonder if such a corrective phase shift would have an influence on the STFT magnitude. The answer is yes. We will demonstrate it.

6.3.2 Invariance of the Fourier magnitude of an analytic signal under a phase shift

We consider a complex temporal function $s(t)$ with magnitude $\rho(t)$ and phase $\theta(t)$:

$$s(t) = \rho(t)e^{i\theta(t)}.$$

Let $F(\omega) = \mathcal{F}(s)(\omega)$ be the Fourier transform of $s(t)$ with magnitude $R(\omega)$ and phase $\alpha(\omega)$:

$$F(\omega) = R(\omega)e^{i\alpha(\omega)} = \int_{\mathbb{R}} \rho(t)e^{i\theta(t)-2i\pi\omega t}dt.$$

It is clear that a phase shift of the form

$$\phi \mapsto s'(t) = \rho(t)e^{i\theta(t)+\phi}$$

does not affect the *magnitude* of the Fourier transform of $s'(t)$ that we call $F'(\omega)$. The constant phase term $e^{i\phi_0}$ just drops out of the integral of the Fourier transform.

Be $\phi \in \mathbb{R}$. It is obvious that

$$\mathcal{F}(s')(\omega) = \int_{\mathbb{R}} \rho(t)e^{i\theta(t)-2i\pi\omega t+i\phi}dt = e^{i\phi}\mathcal{F}(s)(\omega).$$

6.3.3 Invariance of Fourier magnitude of the real part

Under the assumption that s is an analytic function, we get

$$\Im(s) = \mathcal{H}(\Re(s))$$

where $\mathcal{H}$ denotes the Hilbert transform. The following well-known property holds:

$$\mathcal{F}(\Im(s))(\omega) = -i\mathrm{sgn}(\omega)\mathcal{F}(\Re(s)).$$

So we get

$$\begin{aligned}\mathcal{F}(s)(\omega) &= \mathcal{F}(\Re(s))(\omega) + i\mathcal{F}(\Im(s))(\omega)\\ &= (1+\mathrm{sgn}(\omega))\mathcal{F}(\Re(s))(\omega).\end{aligned}$$

Applying this result to $\mathcal{F}(s')$ we get

$$\mathcal{F}(s')(\omega) = 2\chi_{w>0}\mathcal{F}(\Re(s'))(\omega) = e^{i\phi}\mathcal{F}(s)(\omega).$$

So we have the following relation

$$\chi_{w>0}\mathcal{F}(\Re(s'))(\omega) = e^{i\phi}\chi_{w>0}\mathcal{F}(\Re(s))(\omega)$$

that holds for the Fourier transform of a *real* signal. Using the property of hermitian symmetry of the Fourier transform of a real signal, we get

$$\mathcal{F}(\Re(s')) = e^{i\mathrm{sgn}(\omega)\phi}\mathcal{F}(\Re(s)). \tag{6.3}$$

So, a phase shift on the phase of a complex signal does not affect the magnitude of its Fourier transform, and does not affect either the magnitude of the Fourier tranform of its real part, i.e. the signals

$$\begin{cases}\Re(s(t)) = \rho(t)\cos(\theta(t)),\\ \Re(s'(t)) = \rho(t)\cos(\theta(t)+\phi).\end{cases}.$$

both have the same Fourier magnitude.

6.3.4 Study of the consequences of a phase shift on the short-term Fourier transform

We are now considering the STFT operator $\mathcal{S}$:

$$\mathcal{S}(s)(\tau,\omega) = \int_{\mathbb{R}} w(t-\tau)\rho(t)e^{i\theta(t)-2i\pi\omega t}dt$$

We want to know if equation 6.3 still holds when we apply a STFT.

We will see that it is not true, but we will calculate the difference between the STFT squared magnitudes.

We want to know if there exists a relation between $\mathcal{S}(\Re(s')$ and $\mathcal{S}(\Re(s))$, or, at least, between their magnitudes. Remind that $\Re(s') = \Re(s)\cos\phi - \Im(s)\sin\phi$, so

$$S' = S\cos\phi - S_{\mathcal{H}}\sin\phi. \tag{6.4}$$

with

$$\begin{cases}S' = \mathcal{S}(\Re(s')),\\ S = \mathcal{S}(\Re(s)),\\ S_{\mathcal{H}} = \mathcal{S}(\Im(s)) = \mathcal{S}(\mathcal{H}(\Re(s))).\end{cases}$$

So, to understand how a phase shift affects the STFT, we have to understand how it affects the STFT of the Hilbert transform.

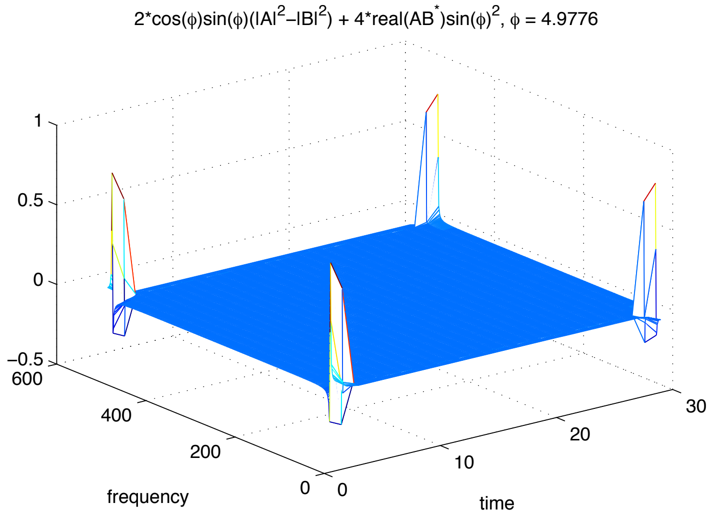

Figure 6.1: STFT squared magnitude error between a pure sine of length $L = 4096$ at $\frac{f}{Fs} = \frac{1000}{44100}$ and the sine shifted of ϕ for STFT using a Hann window of length $N = 512$.

We can note Δ the difference between the squared magnitudes:

$$\Delta = |S|^2 - |S'|^2.$$

So Δ can also be written as a function of ϕ, S and $S_{\mathcal{H}}$:

$$\Delta = (|S|^2 - |S_{\mathcal{H}}|^2)\sin^2\phi + 2\Re(S_{\mathcal{H}}S^*)\cos\phi\sin\phi$$

It is clear that the magnitudes are equal for $\phi \in \{0, \pi\}$ modulo 2π. $\phi = 0$ corresponds to the identity and $\phi = \pi$ corresponds to taking the opposite signal: $s' = -s$. Other interesting values are:

- $\phi = \dfrac{\pi}{2}$: $\quad |S'|^2 = |S_{\mathcal{H}}|^2,$
- $\phi = \dfrac{\pi}{4}$: $\quad |S'|^2 = \frac{1}{2}|S_{\mathcal{H}} - S|^2 = 2\,|B|^2,$
- $\phi = -\dfrac{\pi}{4}$: $\quad |S'|^2 = \frac{1}{2}|S_{\mathcal{H}} + S|^2 = 2\,|A|^2.$

We have calculated the error between the STFT magnitude $|S|^2$ of a real analytic signal and the STFT magnitude $|S'|$ of the same signal shifted in phase. This error is

$$\Delta = (|S|^2 - |S_{\mathcal{H}}|^2)\sin^2\phi + 2\Re(S_{\mathcal{H}}S^*)\cos\phi\sin\phi$$

It includes two terms: one is proportional to the cross-term $\cos\phi\sin\phi$, the other to the quadratic term $\sin^2\phi$. Both terms depend on S and $S_{\mathcal{H}}$ that are that are time-frequency distributions. An example of how the error looks like is presented on figure 6.1: as one can see, the error is small almost everywhere except on the boundaries. On a torus this corresponds to a single point located at both zero time and frequency.

6.4 Open question: temporal phase shift, spectral phase shift

This section sums up several unanswered questions that the author has been asking to himself in order to understand how the short-time Fourier transform affects the phase.

The author first made the wrong assumption that a temporal phase shift of the form

$$x_\phi(t) = \Re((x + (t)i\mathcal{H}\{x\}(t))e^{i\phi} = x(t)\cos(\phi) - \mathcal{H}\{x\}(t)\sin(\phi)$$

led to a spectral phase shift in the time-frequency domain

$$\mathcal{S}\{x_\phi\}(\tau, \omega) = X_\phi(\tau, \omega) \tag{6.5}$$

where we define

$$X_\phi(\tau, \omega) = \mathcal{S}\{x\}(\tau, \omega)e^{-i\mathrm{sgn}(\omega)\phi}.$$

It was nothing more than an analogy with the property of the Fourier transform:

$$\mathcal{F}\{x_\phi\}(\tau, \omega) = e^{-i\mathrm{sgn}(\omega)\phi}\mathcal{F}\{x\}(\omega, \tau).$$

It was then demonstrated that equation 6.5 is false because the difference $|\mathcal{S}\{x_\phi\}|^2 - |\mathcal{S}\{x\}|^2$ was proven to be non-zero.

However, perhaps an interesting result, but still badly understood, is

$$\frac{\|X_\phi - X\|_2^2}{\|X\|_2^2} = \frac{\|x_\phi - x\|_2^2}{\|s\|_2^2} \tag{6.6}$$

for all $\phi \in \mathbb{R}$. So, the temporal phase shift $x \mapsto x_\phi$ has the same SNR than the spectral phase shift $X \mapsto X_\phi$, where x (resp. X) stands for the signal and $x - x_\phi$ (resp. $X - X_\phi$) stands for the noise. The open questions that immediately follow are:

- What is the link between x and X_ϕ ?
- How can one explain the equality 6.6 ?
- Does this equality may help in optimizing the already existing phase retrieval techniques such as Griffin-Lim ?

We now give the demonstration of equation 6.6.

6.4.1 Distance between X and X_ϕ

We are going to calculate the distance between X and X_ϕ using the L^2 norm, such that

$$\|S\|_2^2 = \int_\mathbb{R}\int_\mathbb{R} |X(\tau, \omega)|^2 d\tau d\omega$$

With the assumption that $X_\phi(\tau, \omega) = X(\tau, \omega)e^{-i\mathrm{sgn}(\omega)\phi}$ we calculate the distance $\|X - X_\phi\|$.

$$\|X - X_\phi\|_2^2 = \int_\mathbb{R}\int_\mathbb{R} |1 - e^{-i\mathrm{sgn}(\omega)\phi}|^2 |X(\tau, \omega)|^2 d\tau d\omega$$

$|1 - e^{-i\mathrm{sgn}(\omega)\phi}|^2$ actually does not depend on the sign of ω because $|1 - e^{-i\phi}|^2 = |1 - e^{i\phi}|^2 = 2(1 - \cos\phi)$, so we get

$$\|X - X_\phi\|_2^2 = 2(1 - \cos\phi)\int_\mathbb{R}\int_\mathbb{R} |X(\tau, \omega)|^2 d\tau d\omega = 2(1 - \cos\phi)\|X\|_2^2$$

We can thus express the following spectral signal to noise ratio, where $|1 - e^{i\phi}|$ holds for a noise due to the phase difference between the two STFT.

$$SNR_{spec} = 10\log_{10}\frac{\|S\|_2^2}{\|X - X_\phi\|_2^2} = -10\log_{10} 2(1 - \cos\phi)$$

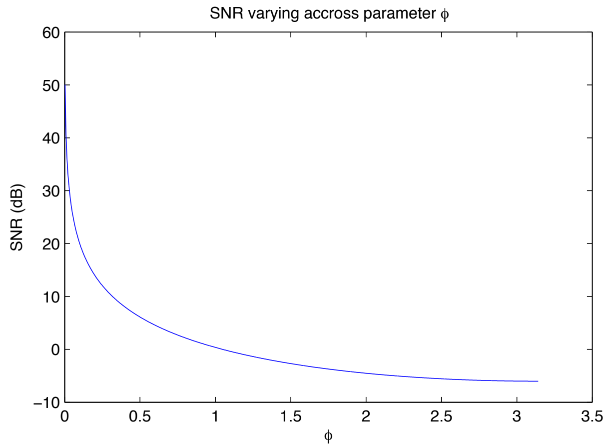

Figure 6.2: Variation of SNR across the phase shift parameter ϕ

6.4.2 Distance between x and x_ϕ

We can also consider the following L^2 distance between the two temporal signals

$$\|x - x_\phi\|_2^2 = \int_{\mathbb{R}} (x(t) - x(t)\cos\phi + \mathcal{H}\{x\}(t)\sin\phi)^2 dt$$

$$= \|x\|_2^2(1-\cos\phi)^2 + \|\mathcal{H}\{x\}\|_2^2 \sin^2\phi + < x, \mathcal{H}\{x\} > (1 - \cos\phi\sin\phi)$$

where $< \cdot, \cdot >$ denotes the usual scalar product on real functions $< x, y >= \int_{\mathbb{R}} x(t)y(t)dt$. We now use the following classical properties of the Hilbert transform.

1. $\|\mathcal{H}\{x\}\|_2 = \|x\|_2$ ($\mathcal{H}$ is an isometry);
2. $< \mathcal{H}\{x\}, x >= 0$ ($\mathcal{H}$ is orthogonal).

We thus find the following expression for the temporal distance

$$\|x - x_\phi\|_2^2 = 2(1-\cos\phi)\|x\|_2^2$$

that allows us to define a temporal signal-to-noise ratio

$$SNR_{temp} = 10\log_{10}\frac{\|x\|_2^2}{\|x - x_\phi\|_2^2} = -10\log_{10} 2(1-\cos\phi).$$

We may notice that $SNR_{spec} = SNR_{temp}$. Theoretically, this signal-to-noise ratio does not depend on the STFT nor on the temporal expression of the signal. It can be seen as a function of the parameter ϕ, as plotted on figure 6.2. It is of course maximal when $\phi = 0$, that is when the two STFT are equal, and minimal for $\phi = \pi$, meaning a complete sign opposition of the two spectrograms and of the corresponding temporal signals.

Conclusion

This report was to study and compare the performances of two phase retrieval algorithms applied to short-time Fourier transform. One is the mainstream Griffin-Lim algorithm that is widely used by signal processing fellowship since its creation in the mid-eighties. The other is a very recent algorithm presented in 2012 in an AES conference by Dennis Sun and Julius Smith. It uses convex optimization techniques to retrieve the phase information from the magnitudes of Gabor frames.

What the phase retrieval algorithms have in common

Those two algorithms use projectors. Griffin-Lim is an alterned projections algorithm. It alternatively projects on:

- the set of time-frequency representations with a given fixed magnitude,
- the set of feasible STFTs, i.e. the result of projector $\Psi^*\Psi$.

Sun-Smith is a projected gradient descent method. That means that at each iteration of the gradient descent, it projects the result on the set of semi-definite positive matrices. Note that this projection needs find the eigenvalues of the matrix xx^T. This operation repeated several times makes the Sun-Smith algorithm very slow.

Differences between the algorithms

The main difference between Griffin-Lim and Sun-Smith is that Sun-Smith is convex while Griffin-Lim is not. The Griffin-Lim algorithm uses the short-time Fourier transform as it was defined and conceptualized in the seventies and eighties: a Fourier transform with a window sliding across time, making a convolution product. The short-term Fourier transform is seen as a time-frequency representation. Indeed, Griffin and Lim wanted to be able to retrieve the phase of a modified STFT for applications such as voice modification. An essential property of the STFT operator is here made use of: the fact that $\Psi\Psi^*$ is projector different from the identity that enable alterning projections.

Sun-Smith sees the short-time Fourier tranform with the more recent approach of a set of scalar products that define time-frequency coefficients. The concept of Hermitian scalar product is here essential because the convex optimization method consists of a different point of view of what can be such a scalar product: instead of a bilinear form on the temporal signal, a linear form on the tensorial product of the temporal signal with itself. The "time-frequency" conception of the STFT is less used, because this is just a slight adaptation of a method that also works on multi-dimensional Fourier transform, and the mathematical properties of the N-D Fourier transform are radically different than the ones of the STFT. The Sun-Smith algorithm is very slow because it considers the problem of phase retrieval from a point of view that is too abstract and does not take sufficient account of the spectral properties of the signal.

Improvement of Griffin-Lim algorithm

A deep study of the Griffin-Lim has been started in this report and the author hopes that it has been clearly explained enough for at least a partial understanding of the objectives. The main problem in the phase retrieval is that for signals with a strong harmonic content, the phase is not random and a bad retrieval of it may lead to suppression of frequencies resulting in a hearing far from what it was aimed to be. It is possible to consider a musical or speech signal as a weighted sum of sines with additive noise, locally stationary, but varying across time. That is why a study on the phase retrieval of a single sine was made. It was observed that Griffin-Lim fails in the retrieval of the phase at the origin. For a single sine on a mono channel, this has no perceptual effects, but it remains both a mathematical and signal processing challenge for the two following reasons:

- although Griffin-Lim algorithm retrieves a signal with STFT magnitude very close to the original, there should be a non-zero difference term due to the phase shift,
- if Griffin-Lim fails in the retrieval of a single sine, it may lead to perceptual alterations in the localisation of sound for stereo channels (or more complex systems) or in the perception of some harmonics.

For those reasons, it could be very interesting to continue those research on improving the phase retrieval of sound from STFT magnitude. The main suggestions for further research proposed by the author are:

- try to find in which way Griffin-Lim retrieves a shifted sine without the difference in STFT magnitude that should be observed,
- optimize the reconstruction of a single sine using that difference of magnitude,
- study the case for superpositions of sines at different frequencies.

The study of the phase shifts on real signals makes use of the Hilbert transform: such a transform is properly defined for continuous signals but its implementation on discrete is not well done yet.

Improvement of Sun-Smith algorithm

As previously said, the Sun-Smith algorithm is very slow because of the several diagonalizations that occur at each iteration. It is actually almost impossible to think of using it on actual audio signals. It is thus necessary to improve the diagonalization, or to find an alternative way of retrieving the signal using convex optimization.

A better understanding of the phases

The phase remains a badly understood concept in signal processing, in particular for a real signal. It could be interesting to use the associated complex analytical signal whose imaginary part is the Hilbert transform of the real part, and this gives interesting theoretical results, but the Hilbert operator is not properly defined yet for discrete signals. There exist connections between the phase of a real signal and the phase of its Fourier transform, or of its short-term Fourier transform. That makes the phase retrieval a delicate problem, that will not be solved until the concept of phases is not more accurately understood by the community.

Appendix A

Tables of comparison between Griffin-Lim and Sun-Smith algorithms

A.1 Random signals with Hann window

The simulations were made on random signals `x = 2*rand(L,1)-1`. It was the same signal for the different values of N, but of course a new signal was generated when changing the size L.

$L = 16$

N = 8	R = 7		R = 6		R = 4		R = 2		R = 0	
	G.-L.	S.-S.	G.-L.	S.-S.	G.-L.	S.-S.	G.-L.	S.-S.	G.-L.	S.-S.
R_1	-6.01e+00	-6.00e+00	-4.45e+00	-6.01e+00	-3.12e+00	9.85e+00	-3.53e+00	7.16e+00	-3.78e+00	3.36e+00
R_2	1.91e+01	1.79e+01	4.46e-01	1.89e+01	9.22e-01	9.89e+00	9.01e-01	8.00e+00	1.45e+00	3.36e+00
R_3	-5.91e+00	-5.99e+00	-5.19e+00	-6.07e+00	-6.01e-01	1.09e+01	-4.07e+00	5.80e+00	-3.70e+00	3.40e+00
R_4	3.23e+02	3.72e+01	3.32e+02	3.56e+01	3.22e+02	1.49e+01	3.41e+02	1.19e+01	Inf	4.80e+00
D_{IS}	0.00e+00	1.42e-02	0.00e+00	1.64e-03	0.00e+00	2.04e+00	0.00e+00	1.10e+00	0.00e+00	2.38e+05
$\frac{1}{v}$	2.69e-02	5.49e-04	6.56e-04	1.12e-03	1.53e-03	2.33e-04	2.23e-01	2.47e-04	5.82e-05	3.66e-04

N = 4	R = 3		R = 3		R = 2		R = 1		R = 0	
	G.-L.	S.-S.	G.-L.	S.-S.	G.-L.	S.-S.	G.-L.	S.-S.	G.-L.	S.-S.
R_1	-4.07e+00	1.58e+01	-4.07e+00	-5.99e+00	-3.90e+00	-5.99e+00	-4.72e+00	-5.67e+00	-4.79e+00	-5.86e+00
R_2	3.00e+00	1.58e+01	3.00e+00	1.58e+01	3.38e+00	1.50e+01	2.17e+00	5.12e+00	2.37e+00	8.31e+00
R_3	-3.91e+00	3.45e+01	-3.91e+00	-6.06e+00	-3.62e+00	-6.07e+00	-3.30e+00	-5.26e+00	-3.35e+00	-6.19e+00
R_4	3.25e+02	3.49e+01	3.25e+02	3.49e+01	3.31e+02	2.70e+01	Inf	1.17e+01	Inf	2.46e+01
D_{IS}	0.00e+00	1.31e-03	0.00e+00	1.31e-03	0.00e+00	7.59e-02	0.00e+00	3.27e- 01	0.00e +00	5.85e-04
$\frac{1}{v}$	4.16e-04	3.90e-04	4.08e-04	4.05e-04	1.13e-03	3.87e-04	8.19e-05	4.25e-04	1.07e-04	5.43e-04

$L = 32$

N = 16	R = 14		R = 12		R = 8		R = 4		R = 1	
	G.-L.	S.-S.	G.-L.	S.-S.	G.-L.	S.-S.	G.-L.	S.-S.	G.-L.	S.-S.
R_1	-1.26e+00	5.63e+00	-1.46e+00	-5.97e+00	-2.33e+00	-4.20e+00	-3.28e+00	-5.03e+00	-3.98e+00	-2.17e+00
R_2	1.03e+00	1.45e+01	-8.33e-01	1.35e+01	2.71e-01	1.49e+00	-1.17e+00	2.88e+00	1.40e+00	1.15e+00
R_3	-1.90e+00	2.55e+01	-1.27e+00	-6.05e+00	-2.62e+00	-4.40e+00	-3.70e+00	-4.96e+00	-4.18e+00	6.87e-01
R_4	3.25e+02	2.64e+01	3.25e+02	3.24e+01	3.24e+02	1.40e+01	3.25e+02	1.21e+01	Inf	1.86e+00
D_{IS}	0.00e+00	5.70e-02	0.00e+00	1.04e-02	0.00e+00	8.49e+00	0.00e+00	5.94e-01	0.00e+00	1.42e+01
$\frac{1}{v}$	3.74e-03	3.54e-04	3.65e-02	4.03e-04	1.15e+00	1.27e-04	4.29e-03	1.39e-04	2.83e-05	3.08e-04

N = 8	R = 7		R = 6		R = 4		R = 2		R = 0	
	G.-L.	S.-S.	G.-L.	S.-S.	G.-L.	S.-S.	G.-L.	S.-S.	G.-L.	S.-S.
R_1	-1.19e+00	2.30e+01	-2.77e+00	1.71e+01	-3.10e-01	-5.13e+00	-1.04e+00	2.82e-01	-1.10e+00	1.67e+00
R_2	7.81e+00	2.30e+01	7.06e+00	1.71e+01	4.48e-01	6.57e+00	2.29e+00	3.95e+00	4.26e+00	3.76e+00
R_3	-1.18e+00	3.72e+01	-2.43e+00	2.55e+01	-5.70e-02	-5.27e+00	-2.88e+00	-1.47e+00	-4.24e+00	-1.02e+00
R_4	3.25e+02	3.76e+01	3.32e+02	2.58e+01	3.26e+02	1.19e+01	Inf	8.86e+00	Inf	9.84e+00
D_{IS}	0.00e+00	2.71e-03	0.00e+00	1.86e-02	0.00e+00	5.97e+00	0.00e+00	6.09e-01	0.00e+00	4.53e-02
$\frac{1}{v}$	1.11e-03	4.30e-04	1.00e-03	5.70e-04	3.43e-03	2.32e-04	1.09e-02	3.58e-04	5.48e-05	4.03e-04

A.2 Sine signals with Hann window

The simulations were made on signals `x = sum(sin(2*pi*f'*(1:L)/Fe),1)` with `f = [6000 7000]` and `Fe = 44100`.

$L = 16$

N = 8	R = 7		R = 6		R = 4		R = 2		R = 0	
	G.-L.	S.-S.	G.-L.	S.-S.	G.-L.	S.-S.	G.-L.	S.-S.	G.-L.	S.-S.
R_1	-5.91e+00	5.61e+00	-1.80e+00	1.43e+01	-8.13e-01	-1.73e+00	-1.88e+00	-5.02e+00	3.85e-01	-1.83e+00
R_2	9.94e+00	1.91e+01	1.53e+00	1.43e+01	1.11e+00	1.57e+00	1.28e+00	3.42e+00	5.78e-01	7.28e-01
R_3	-6.02e+00	2.82e+01	-4.96e+00	2.08e+01	-3.66e+00	-1.56e+00	-2.84e+00	-5.64e+00	-2.18e+00	-2.70e+00
R_4	3.22e+02	2.82e+01	Inf	2.49e+01	3.57e+02	1.69e+01	3.59e+02	1.92e+01	Inf	1.05e +01
D_{IS}	0.00e+00	2.42e-01	0.00e+00	6.66e-01	0.00e+00	3.39e-01	0.00e+00	1.79e-01	0.00e+00	3.13e-01
$\frac{1}{v}$	9.36e-03	4.19e-04	9.53e-04	5.11e-04	1.30e-03	7.17e-04	1.00e-01	4.93e-04	5.46e-05	4.03e-04

N = 4	R = 3		R = 3		R = 2		R = 1		R = 0	
	G.-L.	S.-S.	G.-L.	S.-S.	G.-L.	S.-S.	G.-L.	S.-S.	G.-L.	S.-S.
R_1	-3.41e+00	-5.84e+00	-3.41e+00	7.89e+00	-4.54e+00	-4.68e+00	-3.82e+00	-3.12e+00	-1.77e+00	4.89e+00
R_2	5.91e+00	7.89e+00	5.91e+00	7.89e+00	7.96e-01	2.17e+00	2.32e+00	6.52e+00	2.53e+00	4.89e+00
R_3	-5.19e+00	-6.40e+00	-5.19e+00	2.09e+01	-4.47e+00	-5.15e+00	-3.37e+00	-4.44e+00	-3.20e+00	1.43e+01
R_4	3.30e+02	2.11e+01	3.30e+02	2.09e+01	Inf	1.37e+01	Inf	1.87e+01	Inf	1.43e+01
D_{IS}	0.00e+00	1.12e-02	0.00e+00	1.23e-02	0.00e+00	1.72e-01	0.00e+00	7.61e-03	0.00e+00	9.57e-03
$\frac{1}{v}$	6.45e-04	3.91e-04	6.15e-04	3.91e-04	6.14e-04	3.87e-04	7.96e-05	4.08e-04	1.09e-04	5.54e-04

$L = 32$

N = 16	R = 14		R = 12		R = 8		R = 4		R = 1	
	G.-L.	S.-S.	G.-L.	S.-S.	G.-L.	S.-S.	G.-L.	S.-S.	G.-L.	S.-S.
R_1	1.05e+01	8.01e+00	1.01e+01	3.07e+00	6.60e+00	-4.76e+00	3.02e+00	-1.54e+00	-3.15e-01	-4.66e+00
R_2	1.05e+01	8.01e+00	1.01e+01	4.94e+00	6.72e+00	2.25e+00	4.73e+00	1.46e+00	-2.16e-01	1.99e+00
R_3	7.35e+01	2.48e+01	4.19e+01	5.76e+00	1.31e+01	-5.01e+00	5.16e+00	-1.80e+00	-1.78e+00	-5.47e+00
R_4	3.25e+02	2.53e+01	3.28e+02	1.64e+01	3.44e+02	1.94e+01	3.21e+02	1.72e+01	Inf	1.82e+01
D_{IS}	0.00e+00	3.49e+01	0.00e+00	3.41e+00	0.00e+00	2.05e+00	0.00e+00	1.36e+00	0.00e+00	9.99e-01
$\frac{1}{v}$	5.19e-01	4.61e-04	7.96e-02	2.52e-04	2.72e-01	1.26e-04	1.49e-01	3.27e-04	2.81e-05	4.38e-04

N = 8	R = 7		R = 6		R = 4		R = 2		R = 0	
	G.-L.	S.-S.	G.-L.	S.-S.	G.-L.	S.-S.	G.-L.	S.-S.	G.-L.	S.-S.
R_1	-5.91e+00	-5.39e+00	-1.80e+00	7.81e+00	-8.13e-01	-4.00e+00	-1.88e+00	2.79e-01	3.85e-01	-1.79e+00
R_2	9.94e+00	1.84e+01	1.53e+00	1.17e+01	1.11e+00	1.57e+00	1.28e+00	2.97e+00	5.78e-01	6.09e-01
R_3	-6.02e+00	-5.94e+00	-4.96e+00	1.96e+01	-3.66e+00	-4.87e+00	-2.84e+00	1.08e+00	-2.18e+00	-3.16e+00
R_4	3.22e+02	3.44e+01	Inf	2.35e+01	3.57e+02	1.69e+01	3.59e+02	1.68e+01	Inf	1.34e +01
D_{IS}	0.00e+00	4.79e-01	0.00e+00	9.71e-01	0.00e+00	3.39e-01	0.00e+00	4.32e-01	0.00e+00	7.91e-02
$\frac{1}{v}$	8.62e-03	5.35e-04	8.89e-04	1.10e-03	1.28e-03	5.64e-04	9.89e-02	3.97e-04	5.82e-05	3.98e-04

Appendix B

Matlab codes

B.1 Short-term Fourier transform

```
function X = STFFT(x,w,R)
% Short-term Fourier transform
% x: input signal
% w: analysis window (assume that the order of fft is the length of w)
% R: overlap (in samples)
% w = hann(N,'periodic'); % analysis window

N = length(w);
if(iscolumn(x)==0), x=x'; end
J = length(x);
H = N - R;
M = floor((J-N)/H) + 1;
K = N;
X = zeros(K,M);
for m = 1:M
    deb = (m-1)*H + 1;
    fin = deb + N - 1;
    tx = w.*x(deb:fin);
    X(:,m) = fft(tx,K)/sqrt(K);
end
```

B.2 Inverse short-time Fourier transform

```
function x = ISTFFT(S,w,R)
% Inverse short-term Fourier transform
% S: input spctrogram
% w: synthesis window
% R: overlap (in samples)

[N,M] = size(S);
H = N - R; % hop size
J = (M-1)*H + N;
x = zeros(J ,1);
for m=1:M
    ty = S(:,m);
    ys = ifft(ty,N)*sqrt(N);

    deb = (m-1)*H + 1;
    fin = deb + N - 1;
```

```
    x(deb:fin) = x(deb:fin) + ys .* w; % OLA
end
x = x./(ola(w.^2,H,M)+eps);
```

B.3 Griffin-Lim algorithm

```
function [s,S,conv] = griffin_and_lim(P,w,R,eps)
% [s,S,conv] = griffin_and_lim(P,w,R,eps)
% Computes  signal reconstruction from power spectrogram P
% using Griffin-Lim algorithm.
%
% Input
% P = |S|^2 where S is the STFT of the researched signal
% w: analysis window
% R: size in samples of overlap
% eps: precision for convergence criterion
%
% Output
% s: reconstructed signal
% S: reconstructed spectrogram
% conv: spectral convergence.

S0 = sqrt(P);
S = S0;
ii=1;
if(nargin<4),eps=1e-6;end
obj=1000;
conv = zeros(1,1);
while(obj>eps)
    s = ISTFFT(S,w,R);
    Sr = STFFT(s,w,R);
    S=abs(S0) .* exp(1i*angle(Sr));
    %% Convergence criteria
    SpectralConv_ii = norm(sqrt(P) - abs(Sr))/norm(sqrt(P));
    conv = [conv SpectralConv_ii];
    ii=ii+1;
    obj = abs((conv(ii)-conv(ii-1))/conv(ii-1));
end
s = ISTFFT(S,w,R);
end
```

B.4 Non-convexity of Griffin-Lim algorithm (used in 3.3)

```
% non_convexity_of_griffin_and_lim.m

clear all
close all
clc
% parameters:
L = 2^12;
n = 0:(L-1);
f = 1000;
Fe = 44100;
N = 2^nextpow2(Fe*20e-3);
R = floor(0.75*N);
w = hann(N,'periodic');
% construction of signals:
x = exp(2*1i*pi*f*n'/Fe + 2*1i*pi*rand(1,1));
% boundary conditions:
% x(1:N/2) = x(1:N/2).*w(1:N/2);
% x(end-N/2+1:end) = x(end-N/2+1:end).*w(end-N/2+1:end);
x(1:N/2) = zeros(size(x(1:N/2)));
x(end-N/2+1:end) = zeros(size(x(end-N/2+1:end)));
X1 = abs(STFFT(x,w,R)).^2;
% retrieval of temporal signal from X using Griffin-Lim algorithm:
y = griffin_and_lim(X1,w,R,10^-6);
x = x(1:length(y));
vec = N+1:L-N;
x = x(vec);
y = y(vec);
y = y/max(abs(y));
phi1 = angle(x(1));
phi2 = angle(y(1));
deltaphi = -(phi2 - phi1);
z = exp(1i*deltaphi)*y;
z = z/max(abs(z));
phi3 = angle(z(1));
% computation of signal-to-noise ratios
R1 = SNR(x,x-y);
R2 = SNR(x,x-z);
% plots
figure, plot(real(x(1:500)))
hold on, plot(real(y(1:500)),'r'), plot(real(z(1:500)),'g')
legend(['x, \phi_x = ' num2str(phi1)],...
['y, \phi_y = ' num2str(phi2)],['z, \phi_z = ' num2str(phi3)])
title('Error in the phase at origin for Griffin-Lim retrieval of a sine')
ylim([-1.5 1.5]), xlabel('time (samples)')
```

```
% non_convexity_of_griffin_and_lim_2.m
clear all
close all
clc
% parameters:
L = 2^12;
n = 0:(L-1);
f = 1000;
Fe = 44100;
N = 2^nextpow2(Fe*20e-3);
R = floor(0.75*N);
w = hann(N,'periodic');
% construction of signals:

Ntours = 50;
phi_x = zeros(Ntours,1);
phi_y = zeros(Ntours,1);
alpha = linspace(0,1,Ntours);

for ii = 1:Ntours
x = real(exp(2*1i*pi*f*n'/Fe + 2*1i*pi*alpha(ii)));
% boundary conditions:
% x(1:N/2) = x(1:N/2).*w(1:N/2);
% x(end-N/2+1:end) = x(end-N/2+1:end).*w(end-N/2+1:end);
x(1:N/2) = zeros(size(x(1:N/2)));
x(end-N/2+1:end) = zeros(size(x(end-N/2+1:end)));
X1 = abs(STFFT(x,w,R)).^2;
% retrieval of temporal signal from X using Griffin-Lim algorithm:
y = real(griffin_and_lim(X1,w,R,10^-6));
x = x(1:length(y));
vec = N+1:L-N;
x = x(vec);
y = y(vec);
y = y/max(abs(y));
phi1 = acos(x(1));
phi2 = acos(y(1));
% computation of signal-to-noise ratios
R1 = SNR(x,x-y);
% plots
figure(1), plot(real(x(1:500)))
hold on, plot(real(y(1:500)),'r')
legend(['x, \phi_x = ' num2str(phi1)],...
['y, \phi_y = ' num2str(phi2)])
title('Error in the phase at origin for Griffin-Lim retrieval of a sine')
ylim([-1.5 1.5]), xlabel('time (samples)')
set(gcf, 'papersize', [15 10]);set(gcf, 'paperposition', [0 0 15 10]);
print -dpdf non_convexity.pdf
phi_x(ii) = phi1;
phi_y(ii) = phi2;
end

figure(2),plot(phi_x,phi_y)
figure(2), xlabel('\phi_x'), ylabel('phi_y')

figure(3), plot(phi_x), hold on, plot(phi_y,'r')
xlabel('No. of iteration'),ylabel('Phase at origin')
legend('original signal x','retrieved signal y')
```

Appendix C

The Hilbert transform of continuous signals

C.1 Definitions

Definition C.1.1 (Integral definition)**.** Let $x \in L^2(\mathbb{R})$. The Hilbert transform of x is noted $\hat{x}$ and defined by

$$\hat{x}(t) = \frac{1}{\pi} p.v. \int_{\mathbb{R}} \frac{x(\tau)}{t-\tau} d\tau$$

where $p.v.$ denotes the Cauchy principal value.

It may be convenient to see the Hilbert transform of x as the convolution with the Cauchy principal value of $\frac{1}{\pi t}$ distribution.

Proposition C.1.1 (Convolution definition)**.** *Let $x \in L^2(\mathbb{R})$. The Hilbert transform of x is given by the convolution product*

$$\hat{x} = x * p.v. \left(\frac{1}{\pi t}\right)$$

Proposition C.1.2. *Let f be a complex function analytic in the plane $\{\Re z > 0\}$. Then f can be written*

$$f = x + i\hat{x}$$

where $x = \Re f$ and $\hat{x} = \Im f$.

Definition C.1.2 (Analytic signal)**.** Let $x \in L^2(\mathbb{R})$ be a real-valued signal. The analytic signal associated to x is defined as $f = x + i\hat{x}$.

C.2 The Hilbert operator

Definition C.2.1 (Hilbert operator)**.** The Hilbert operator is defined as

$$\mathcal{H} : \left| \begin{array}{ccc} L^2(\mathbb{R}) & \longrightarrow & L^2(\mathbb{R}) \\ x & \longmapsto & \hat{x} \end{array} \right.$$

Proposition C.2.1 (Properties of the Hilbert operator)**.** *The Hilbert operator has the following properties:*

1. $\mathcal{H}^2 = -Id$;
2. $\mathcal{H}^3 = -\mathcal{H}$;
3. $\mathcal{H}^4 = Id$.

where Id is the identity operator and the power means the composition. It follows that $\mathcal{H}$ is invertible and $\mathcal{H}^{-1} = -\mathcal{H}$.

Proposition C.2.2 (Hilbert transform of usual functions and distributions [Weisstein,])**.** *We give below the Hilbert transforms of some usual functions. ω is a real constant.*

$x(t)$	$\hat{x}(t)$
$\cos(\omega t)$	$\sin(\omega t)$
$\sin(\omega t)$	$-\cos(\omega t)$
e^{it}	$-ie^{it}$
e^{-it}	ie^{-it}
e^{-t^2}	$e^{-t^2}\int_0^t e^{\tau^2} d\tau$

This property leads to the eigenvalue characterization of the Hilbert operator.

Theorem C.2.1. *The Hilbert operator $\mathcal{H}$ has two eigenvalues that are i and $-i$. The associated eigenvectors are respectively $\{t \mapsto e^{-it}\}$ and $\{t \mapsto e^{it}\}$.*

Proposition C.2.3 (Hilbert transform of periodic functions [Pandey, 1995])**.** *If x is a T-periodic function, its Hilbert transform is defined by*

$$\hat{x}(t) = \frac{1}{2T}\ p.v. \int_{-T}^{T} x(t-\tau)\cot\left(\frac{\tau\pi}{2T}\right) d\tau$$

C.3 Links with Fourier transforms

Theorem C.3.1 (Spectral properties of the Hilbert transform)**.** *$\mathcal{F}$, $\mathcal{H}$ and $\mathcal{S}$ respectively denote the Fourier, Hilbert and STFT operators.*

1. *Let $x \in L^2(\mathbb{R})$ be a real-valued function that admits a Hilbert transform.*

   $$\mathcal{F}\{\mathcal{H}\{x\}\}(\omega) = -i \cdot \operatorname{sgn}(\omega)\mathcal{F}\{x\}(\omega)$$

   where $\operatorname{sgn}(\omega)$ *is the sign of* ω*.*

2. *If we call $f = x + i\mathcal{H}x$ the analytical signal associated to x, we have*

   $$F\{f\}(\omega) = (1+\operatorname{sgn}(\omega))f(\omega) = \begin{cases} 0 \ if\ \omega < 0, \\ f(0)\ if\ \omega = 0, \\ 2f(\omega)\ if\ \omega > 0 \end{cases}$$

Proposition C.3.1. *Let $x \in L^2(\mathbb{R})$ be a real-valued function that admits a Hilbert transform $\hat{x}$. The following equalities hold.*

1. $x * x + \hat{x} * \hat{x} = 0$ *or equivalently*
2. $F\{x\}^2 + F\{\hat{x}\}^2 = 0$;
3. $F\{x^2\} + F\{(\hat{x})^2\} = 0$ *or equivalently*
4. $F\{x\} * F\{x\} + F\{\hat{x}\} * F\{\hat{x}\} = 0$.

Bibliography

[Allen and Rabiner, 1977] Allen, J. B. and Rabiner, L. R. (1977). A unified approach to short-time fourier analysis and synthesis. *Proceedings of the IEEE*, vol. 65, no. 11.

[Bouvrie and Ezzat, 2006] Bouvrie, J. and Ezzat, T. (2006). An incremental algorithm for signal reconstruction from stft magnitude. In *ICSLP, Pittsburgh, PA, USA*.

[Boyd and Vandenberghe, 2004] Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization*.

[Candès et al., 2011] Candès, E. J., Th., S., and V., V. (2011). Phaselift: Exact and stable signal recovery from magnitude measurements via convex programming. *ArXiV e-prints*.

[Capricelli, 2008] Capricelli, T. D. (2008). Algorithmes de projections convexes généralisées et applications en imagerie médicale. Thèse de l'Université Pierre et Marie Curie.

[Combettes and Pesquet, 2010] Combettes, P. L. and Pesquet, J.-C. (2010). Proximal splitting methods in signal processing. *IEEE*.

[Cooley and Tukey, 1965] Cooley, J. W. and Tukey, J. W. (1965). An algorithm for the machine calculation of complex fourier series. *Mathematics of Computation*, Vol. 19, No. 90.

[Fienup, 1982] Fienup, R. J. (1982). Phase retrieval algorithm: a comparison. *Applied Optics*, Vol. 21, No. 15.

[Flanagan and Golden, 1966] Flanagan, J. L. and Golden, R. (1966). Phase vocoder. *Bell Syst. Tech. J.*, vol. 45.

[Gabor, 1946] Gabor, D. (1946). Theory of communication. *Institution of Electrical Engineering*.

[Gerchberg and Saxton, 1971] Gerchberg, R. W. and Saxton, W. O. (1971). A practical algorithm for the diffraction of phase from image and diffraction plane pictures. *Optik*, Vol. 35 (No. 2).

[Gold et al., 1969] Gold, B., Oppenheim, A. V., and Rader, C. M. (1969). Theory and implementation of the discrete hilbert transform. *Symposium on computer processing in communication*.

[Griffin and Lim, 1984] Griffin, D. W. and Lim, J. S. (1984). Signal Estimation from Modified Short-Time Fourier Transform. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. ASSP-32, No. 2.

[Hiriart-Urruty, 1978] Hiriart-Urruty, J.-B. (1978). On optimality conditions in nondifferentiable programming. *Mathematical Programming*.

[Kaiblinger, 2005] Kaiblinger, N. (2005). Approximation of the fourier transform and the dual gabor window. *Journal of Fourier Analysis and Application*, vol. 11.

[Mallat, 2008] Mallat, S. (2008). *A Wavelet Tour of Signal Proceesing*, chapter 4: Time meets frequency, page 63. `http://www.cmap.polytechnique.fr/~mallat/papiers/WaveletTourChap1-2-3.pdf`.

[Mitchell, 2000] Mitchell, J. E. (2000). solving semidefinite programming problems using branch-and-bound. In *IPM 2000 and EURO 2000, Budapest*.

[Moreau, 1965] Moreau, J.-J. (1965). Proximité et dualité dans un espace hilbertien. *Bull. Soc. math. France.*

[Pandey, 1995] Pandey, J. (1995). The hilbert transform of periodic distributions. *Integral Transforms and Special Functions*, 5.

[Portnoff, 1976] Portnoff, M. R. (1976). Implementation of the digital phase vocoder using the fast fourier transform. *IEEE Trans. Acoustics, Speech, Signal Proc.*, vol. ASSP-24, no. 3.

[Portnoff, 1978] Portnoff, M. R. (1978). *Time-scale modification of speech based on short-time Fourier analysis.* PhD thesis, Massachussets Institute of Technology.

[Quatieri and Oppenheim, 1981] Quatieri, T. F. J. and Oppenheim, A. V. (1981). Iterative techniques for minimum phase signal reconstruction from phase or magnitude. *IEEE Transactions on Acoustics, Speech and Signal Processing*, ASSP-29, No. 6.

[Recht, 2012] Recht, B. (2012). Projected gradient methods. Available at `http://pages.cs.wisc.edu/~brecht/cs726docs/ProjectedGradientMethods.pdf`.

[Sturmel and Daudet, 2011] Sturmel, N. and Daudet, L. (2011). Signal reconstruction from STFT magnitude: a state of the art. In *International Conference on Digital Audio Effect (DAFx-11).*

[Sun and Smith, 2012] Sun, D. L. and Smith, J. O. (2012). Estimating a signal from a magnitude spectrogram via convex optimization. In *133rd Convention of the Audio Engineering Society.* `http://arxiv.org/abs/1209.2076`.

[Voroninski, 2012] Voroninski, V. (2012). A comparison between the PhaseLift and PhaseCut algorithms. Available at `http://math.berkeley.edu/~vladv/PhaseCutProofs.pdf`.

[Waldspurger et al., 2012] Waldspurger, I., d'Aspremont, A., and Mallat, S. (2012). Phase Recovery, MaxCut and Complex Semidefinite Programming. *ArXiv e-prints.* `http://arxiv.org/abs/1206.0102`.

[Weisstein,] Weisstein, E. W. "Hilbert Transform." From MathWorld–A Wolfram Web Resource. `http://mathworld.wolfram.com/HilbertTransform.html`.

[Zhechev, 2005] Zhechev, B. (2005). Hilbert transform relations. *Cybernetics and information technologies.*