Harmonic Sinusoid Modeling of Tonal Music Events

Wen, Xue

For additional information about this publication click this link.
http://qmro.qmul.ac.uk/xmlui/handle/123456789/15043

Information about this research object was correct at the time of download; we occasionally make corrections to records, please therefore check the published record when citing. For more information contact scholarlycommunications@qmul.ac.uk

Harmonic Sinusoid Modeling of Tonal Music Events

Wen, Xue

Centre for Digital Music
Department of Electronic Engineering
Queen Mary, University of London

A thesis submitted for the degree of
Doctor of Philosophy
of the University of London

October, 2007

“I certify that this thesis and the research to which it refers are the product of my own work, and that any ideas or quotations from the work of other people, published or otherwise, are fully acknowledged in accordance with the standard referencing practices of the discipline.

Wen, Xue

Abstract

This thesis presents the theory, implementation and applications of the harmonic sinusoid modeling of pitched audio events.

Harmonic sinusoid modeling is a parametric model that expresses an audio signal, or part of an audio signal, as the linear combination of concurrent slow-varying sinusoids, grouped together under harmonic frequency constraints. The harmonic sinusoid modeling is an extension of the sinusoid modeling, with the additional frequency constraints so that it is capable to directly model tonal sounds. This enables applications such as object-oriented audio manipulations, polyphonic transcription, instrument/singer recognition with background music, etc.

The modeling system consists of an analyzer and a synthesizer. The analyzer extracts harmonic sinusoidal parameters from an audio waveform, while the synthesizer rebuilds an audio waveform from these parameters. Parameter estimation is based on a detecting-grouping-tracking framework. The detecting stage finds and estimates sinusoid atoms; the grouping stage collects concurrent atoms into harmonic groups; the tracking stage collects the atom groups at different time to form continuous harmonic sinusoid tracks. Compared to standard sinusoid model, the harmonic model focuses on harmonic groups of atoms rather than on isolated atoms, therefore naturally represents tonal sounds. The synthesizer rebuilds the audio signal by interpolating measured parameters along the found tracks.

We propose the first application of the harmonic sinusoid model in digital audio editors. For audio editing, with the tonal events directly represented by a parametric model, we can implement standard audio editing functionalities on tonal events embedded in an audio signal, or invent new sound effects based on the model parameters themselves. Possibilities for other applications are suggested at the end of this thesis.

Acknowledgements

The past three years have been a most special period of my studying life, not only for being the last stage of my formal education, but also for being my first experience of doing research in a foreign country, within a culture so different from my own, and in a course for the first time being 100% my free choice. The final arrival at this thesis has been possible only with the immense support I received from my family, colleagues and fellow researchers, to whom I am most obliged, and to whom I dedicate my acknowledgements.

Thanks to both my parents, who, while combating the greatest difficulties of life, have been supporting me to continue my study in a faraway land. And also to my dear brother, who has undertaken my family obligations as his own.

Thanks to my supervisor Prof. Mark Sandler for recruiting me into this area and this wonderful Centre for Digital Music, for the continuous support he has given me all through the years, for the valuable advices on my research works, and above all, for making all these happen.

Thanks to the European Commission, the Higher Education Funding Council for England, and Queen Mary, University of London for their kind financial support during my study.

Thanks to Dr. Juan Bello, who introduced me to a variety of topics at the start of my study, and who had been in frequent discussion of technical issues.

Thanks to Dr. Mark Plumbley and Dr. Joshua Reiss for their suggestions and comments that have accompanied all the stages of my study. To Lynda Rolfe and other staff members in the department for the non-technical helps I get from them.

Thanks to fellow researchers Anssi Klapuri, Sylvain Marchand and Simon Dixon for their suggestions on technical issues. To Dan Ellis for suggesting C4DM as the place for my PhD. To Adam Berenzweig for sharing dataset.

Thank my dear colleagues and friends, Amélie, Andrew, Andrew, Becky, Beiming, Chris, Chris, Chris, Chris, Chris, Dan, Daohui, Dawn, Emmanuel, Enrique, George, Giuliano, Katy, Kurt, Lauren, Louis, Maria, Mark, Matthew, Max, Matthias, Nico, Panos, Paul, Peyman, Ranting, Samer, Steve, Thomas, Yves, for their various supports, advices, discussions, cakes, and all the good times we are together.

.

Table of contents

List of figures

List of tables

List of variables

Variable	As	Stands for
a		Amplitude
b		Another amplitude, following a
B		Stiffness coefficient
F		Frequency
$F0$		Pitch
g		Another frequency, following f
k	subscript superscript	1) Frequency bin index (discrete) 2) Frequency, measured in bins (continuous) 3) Event index
K		Number of events
l	subscript	Frame index
L	subscript	Number of frames
m	superscript	Partial index
n		Time, in samples (discrete)
N		1) Size of analysis window 2) Size of audio excerpt 3) Number of vertices of a polygon
s		Score
r		Residue
t		Time (continuous)
T		Sampling period
w		Window function
θ		1) Another phase angle, following φ 2) A value between 0 and 1
τ		Decay time constant
φ		Phase angle

List of symbols

x	A variable or a signal
$\hat{x}$	Estimate of x
$\tilde{x}$	Resynthesized x
$\underline{x}$	High-pass approximation of x
$\bar{x}$	Low-pass approximation of x
x^m	The mth partial of x
x_n	A discrete-time signal x regarding time n
(x_n)	The sequence of x_n indexed on n
$x(t)$	A continuous-time signal x regarding time t
$X(f)$	The discrete-time Fourier transform of x, f being the digital frequency
X_k	The discrete Fourier transform of x, k being the frequency bin index
Δx	1) An increment of x 2) Difference of x
$\Delta^k x$	kth-order difference of x

Introduction

Musical signal processing is often categorized into two sub-areas: low-level and high-level processing, depending on the use of symbolic representations (scores). Traditionally, high-level processing focuses on symbolic data, while low-level processing focuses on audio. The *musical note*, which is the basic element of the symbolic system, functions to communicate between the two levels. On one hand, many operations that generate symbolic data from audio, such as onset detection [BDADDS05], musical instrument recognition [ERD05, EK00], pitch estimation [Klapuri99, DG02], etc., are defined for individual notes or note groups rather than for arbitrary audio content; on the other hand, the operations that generate audio from symbolic data, known as music synthesis [Howe75], almost always proceed note by note. The symbolic-level representation for musical notes is already well defined. However, to fulfil its role as the bridge between low and high levels, we still need a representation for musical notes on the low level.

The motivation of harmonic sinusoid modeling for music is that a large number of musical notes are pitched. The term *pitch* is defined as "the attribute of auditory sensation that orders sounds on a scale extending from low to high" [ASA60]. In music, the pitch is a basic property of musical notes, together with other properties such as duration, loudness and timbre [FR98]. In melodic music, a melody line is composed of a group of pitch values attached to a sequence of time intervals. Orchestral instruments, such as the strings and the winds, are good examples of pitched sound sources. Depending on the mechanism of generation, the string sounds can be categorized into plucked (e.g. guitar), struck (e.g. piano) or bowed (e.g. the viols), the wind instrument sound into lip driven (e.g. the brass), reed driven (e.g. clarinet) or air flow driven (e.g. flute, organ). The human voice, which can involve extremely complicated mechanisms, is often pitched during vocal music performance. Some mallet percussion instruments (marimba, etc.) are also perceived as pitched. Modern electrical instruments can deliver any sound in theory. However, when they are used to play a melody, the sounds are usually well pitched.

The pitch is often closely related to *periodicity* [Hartmann96]. For audio signals, periodicity refers to the periodic behaviour of the time-domain waveform. A key descriptor of periodicity is the *period*. Signals with perfect periodicity are invariant to the time shift of a period. *Harmonicity* is the frequency-domain counterpart of periodicity. A key descriptor of harmonicity is the *fundamental frequency*. Signals with perfect harmonicity have their frequency-domain energy concentrated at multiples of the fundamental. Numerically the fundamental frequency is reciprocal to the period. The equivalence of periodicity and harmonicity are shown by the harmonic decomposition known as *Fourier series*. Using Fourier series a periodic waveform is expressed as the linear combination of *harmonic sinusoids*, i.e. the frequencies of all the sinusoids are multiples of a fundamental frequency. We call this the *harmonic sinusoid representation* of the periodic signal.

The use of harmonic sinusoids for representing pitched musical notes is supported by musical acoustics studies [FR98]. According to these studies, harmonic sinusoids are the steady-state response of a 1-dimension simple oscillating system. Examples of such systems include strings (e.g. piano, viols) and air columns (e.g. organ, woodwinds). Another source of harmonic sinusoids is periodic stimulus, of which the human voice is an example. The studies also show that musical instruments with other oscillation mechanisms can produce non-harmonic sinusoids. Mallet-bar percussions, such as the marimba, are examples of "pitched" instruments using 2-dimensional oscillating bodies. These instruments show good harmonicity in high partials, but have significant *inharmonicity* in the low (and usually strong) partials. In this thesis the harmonic sinusoid representation is only applied to those pitched sounds that do have a harmonic or quasi-harmonic structure. We use the symbol $F0$ to refer to the pitch, measured in the same dimension as frequency. A sound with pitch $F0$ is perceived as having the same position on the pitch scale as a sinusoid with frequency $F0$ has, no matter if it has a fundamental frequency, or if its fundamental frequency equals $F0$.

The harmonic sinusoid representation provides a compact way for describing periodic signals. However, the harmonic sinusoid modeling of real-world pitched notes cannot be accomplished using the Fourier series. The main reason is that real-world notes are not strictly periodic: it is common for them to have changes from one period to the next. This motivates the use of time-varying sinusoids in the harmonic sinusoid representation. Parameters of time-varying sinusoids are evaluated locally as functions of time. Time-varying sinusoids have been successfully used in standard sinusoid modeling [MQ86, Serra89] without harmonic context. To use time-varying sinusoids for modeling pitched notes, special care should be taken regarding the harmonicity between sinusoids. This is the starting point of upgrading sinusoid modeling to harmonic sinusoid modeling, the latter being the focus of this thesis.

7 articles written during this PhD course have been published or accepted for publish, as listed below.

[WS05] Wen X. and M. Sandler, "Transcribing piano music using signal novelty," in *Proc. AES 118th Convention*, Barcelona, 2005.

This article describes the improving of music transcription by suppressing the spectral contribution of previous events before estimating the pitch of a new note. After the suppression, the target spectrum gains better harmonicity corresponding to the new event. This method involves no explicit sinusoid model and is not discussed in this thesis.

[WS05b] Wen X. and M. Sandler, "A partial searching algorithm and its application for polyphonic transcription," in *Proc. ISMIR'05*, London, 2005.

This article proposes a partial searching algorithm that models inharmonicity by adaptively allowing frequency departure from the perfect harmonic position. The algorithm is embedded in a polyphonic music transcription system, using the pitch hypotheses as a clue. This inharmonicity model is not so robust as the one in §3.2, and is not discussed in this thesis.

[WS06] Wen X. and M. Sandler, "Error compensation in modeling time-varying sinusoids," in *Proc. DAFx'06*, Montreal, 2006.

This article describes a method for re-estimating sinusoidal parameters using parameter dynamics information embedded in sinusoid tracks. It is further developed in §3.3 and Appendix E.2. This method is a part of the harmonic sinusoid analyzer. Due to its high computational cost, it is likely to be replaced by the more efficient method in Appendix E.3 in future systems.

[WS07] Wen X. and M. Sandler, "New audio editor functionality using harmonic sinusoids," in *Proc. AES122nd Convention*, Vienna, 2007.

This article introduces new audio editor operations enabled by the harmonic sinusoid model. It includes an introduction to the model and the involved techniques, and details on the interface design issues and audio editing operations. Examples are included in Chapter 5 of this thesis.

[WS07b] Wen X. and M. Sandler, "Sinusoid modeling in a harmonic context," in *Proc. DAFx'07*, Bordeaux, 2007.

This article is a miniature version of this thesis. It explains the theory and implementation of harmonic sinusoid modeling, including most key points in Chapters 3 and 4, along with selected numerical results.

[WS07c] Wen X. and M. Sandler, "Calculation of radix-2 discrete multiresolution Fourier transform," *Signal Processing*, vol. 87 no.10, 2007, pp.2455-2460.

This article discusses the calculation of radix-2 discrete multiresolution Fourier transforms (DMFT). The DMFT is a redundant time-frequency-scale representation that includes DFT's calculated using multiple window sizes. In this article we show how we can compute a radix-2 DMFT saving up to 50% computation. It is only marginal related to sinusoid modeling, and is not discussed in this thesis.

[WS07d] Wen X. and M. Sandler, "A composite spectrogram using multiple Fourier transforms," to appear in *IET Signal Processing*.

This article introduces a DMFT-based time-frequency representation that automatically chooses windows sizes for individual areas in the time-frequency plane, so that the result is optimized in e.g. the minimal entropy sense. It also includes a fast algorithm for the selection. This topic is not discussed in this thesis.

The main chapters of this thesis are arranged as follows.

Chapter 1 defines the harmonic sinusoid model. Examples of harmonic sinusoids are provided with contexts in musical acoustics or signal processing. The spectral properties of stationary and slow-varying sinusoids are discussed. The structure of harmonic sinusoids in the modeling system is also presented.

Chapter 2 provides a brief review of techniques involved in standard sinusoid modeling, including additional discussions of the error bounds in DFT-based parameter estimators. This chapter also shows how we develop harmonic sinusoid modeling from standard sinusoid modeling, giving a comparison between concepts and components of the two.

Chapter 3 discusses the estimation of sinusoidal parameters and the grouping of sinusoid atoms by harmonicity. The least-square-error estimator is presented with emphasis on frequency-domain implementation. Harmonic grouping is discussed with an inequality-based harmonic model, designed to tolerate frequency estimation errors. Two techniques for re-estimating parameters from time-varying sinusoid tracks are also presented.

Chapter 4 discusses the harmonic tracking based on the same inequality model. Topics of this chapter include tracking criteria based on frequency and amplitude continuity, forward tracking of single/multiple harmonic sinusoids, forward-backward tracking of single harmonic sinusoid, as well as terminating conditions.

Chapter 5 discusses the applications of harmonic sinusoid models. We present the implementation of new audio editor operations in details, and briefly outline other possible applications. The harmonic synthesis technique for reconstructing time-domain harmonic sinusoids from model parameters is discussed at the beginning of this chapter.

Chapter 6 summarizes the whole thesis, discusses its research contributions, and proposes further direction to improve harmonic sinusoid modeling.

This thesis also includes 6 appendices containing definitions, mathematical proofs, algorithms and computation details.

The main contributions of this thesis include:

- The harmonic sinusoid modeling system (§2.6).
- Robust representation of harmonic frequency contents and its application for finding harmonic signal components from the spectrum (§3.2).
- Joint operation of harmonic grouping and harmonic tracking (§4.1~§4.6).
- Estimating sinusoids using the knowledge of signal dynamics (§3.3, §3.4).
- Application of harmonic sinusoids for audio editing (§5.2).

A more detailed list of contribution points, including minor ones, will be given in §6.2.

Chapter 1

Sinusoids and harmonic sinusoids

This chapter is devoted to the definition of harmonic sinusoids and studying the behaviour of time-varying sinusoids. In 1.1 we define the harmonic sinusoid model. Examples of harmonic sinusoids are provided in 1.2 with references to music acoustics. 1.3 discusses the Fourier transform, on which most of the thesis work is based, and applies it to time-varying sinusoids. 1.4 discusses the uniqueness issue and how it is related to the slowness of parameter variations. 1.5 defines the harmonic sinusoid modeling system, and explains the structure of harmonic sinusoid representation within this system.

1.1 Harmonic sinusoids

The *harmonic sinusoid model* represents a pitched music event as the linear combination of a number of *partials*, each partial being a slow-varying sinusoid, as follows:

$$x(t)=\sum_{m=1}^{M}x^m(t)=\sum_{m=1}^{M}a^m(t)\cos\varphi^m(t) \qquad (1.\ 1a)$$

where M is the number of partials, and $x^m(t)$ the m^{th} partial. $a^m(t)$ is a slow-varying function of t, interpreted as the *instantaneous amplitude* of the m^{th} partial. $\varphi^m(t)$ is interpreted as the phase angle, with its derivative (divided by 2π), $f^m(t)$, being another slow-varying function of t, known as the *instantaneous frequency*:

$$f^m(t)=\frac{d\varphi^m(t)}{dt},\quad \varphi^m(t)=\varphi^m(0)+2\pi\int_0^t f^m(t)dt \qquad (1.\ 1b)$$

In sinusoid modeling, a signal in the form specified by (1.1a) and (1.1b) is often known as *deterministic*, implying it has stable evolution properties. However, it does not necessarily have a fundamental frequency. In harmonic sinusoid model we impose the constraint that the *M* partials are harmonic or quasi-harmonic. Perfect harmonicity is expressed as

$$f^m(t) = mf^1(t) \tag{1. 1c}$$

i.e. all partial frequencies are multiples of the lowest frequency, known as the *fundamental frequency*. In practice we also see the phenomenon of inharmonicity, i.e. pitched sounds diverge from perfect harmonicity. This is expressed by

$$f^m(t) = mF0(t) + \delta f^m(F0(t)) \tag{1. 1d}$$

where $\delta f^m(F0)$ is the frequency departure of the m^{th} partial from perfect harmonicity. Here we have used F0 instead of f^1 to indicate that the lowest partial is allowed to have a frequency departure too.

Equations (1.1a)~(1.1d) define harmonic sinusoids as continuous signals. We derive the discrete version by sampling $x(t)$, $x^m(t)$, $a^m(t)$ and $\varphi^m(t)$ at multiples of a *sampling period* T, i.e.

$$x_n = x(t)\,|_{t=nT}, \quad a_n = a(t)\,|_{t=nT}, \quad \varphi_n = \varphi(t)\,|_{t=nT} \tag{1. 2}$$

Then the discrete versions of (1.1a) and (1.1b) are obtained as

$$x_n = \sum_{m=1}^{M} x_n^m = \sum_{m=1}^{M} a_n^m \cos\varphi_n^m, \quad \varphi_n^m = \varphi_0^m + 2\pi\int_0^{nT} f^m(t)dt \tag{1. 3}$$

Up to now we have not associated the time variable *t* with a unit. It is convenient to choose this unit to be the sampling period, i.e. T=1. Then $x_n = x(t)\,|_{t=n\cdot 1} = x(n)$, $a_n^m = a^m(n)$, $\varphi_n^m = \varphi^m(n)$, and (1.3) becomes

$$x_n = \sum_{m=1}^{M} x_n^m = \sum_{m=1}^{M} a_n^m \cos\varphi_n^m, \quad \varphi_n^m = \varphi_0^m + 2\pi\int_0^{n} f^m(t)dt \tag{1. 4}$$

Equation (1.4), together with (1.1d), defines discrete harmonic sinusoids. In deriving (1.4) we have ignored the sampling alias. This approximation is valid only if all partial frequencies are well below the *Nyquist frequency*, i.e. half the sampling frequency. For most musical materials sampled at 44.1kHz, this condition is satisfied during the process of recording and digital transfer. The instantaneous frequency

$f^m(t)$ is not discretized in (1.4). In fact, $f^m(t)$ affects x_n only by its integrals over intervals that start and end at the sampling points. We define

$$\Delta\varphi_n^m = 2\pi\int_n^{n+1} f^m(t)dt \tag{1. 5}$$

then

$$\Delta\varphi_n^m = \varphi_{n+1}^m - \varphi_n^m, \quad \varphi_n^m = \varphi_0^m + \sum_{l=0}^{n-1}\Delta\varphi_l^m \tag{1. 6}$$

$\Delta\varphi_n^m$ is interpreted as the linear average of the instantaneous frequency on the interval $[n, n+1]$. Again, a_n^m and $\Delta\varphi_n^m$ are slow-varying functions of n.

Harmonic sinusoids are closely related to periodicity in time domain. If we lift the time dependency of a^m and f^m and let $f^m = mf^1$, then (1.1a) is reduced to a Fourier series, which exactly represents a periodical signal. The period is the reciprocal of the fundamental frequency f^1. Slow variations of a^m and f^m introduce small variations in the wave shape from one period to the next. The inharmonicity introduces dispersions to the waveform.

In music audio a harmonic sinusoid defined above models a single music note or a sequence of notes connected by smooth pitch variation. If there is an abrupt change between consecutive notes, or if the music has more than one concurrent notes, it is modeled as a combination of harmonic sinusoids:

$$x_n = \sum_{k=1}^{K} x_n^k = \sum_{k=1}^{K}\sum_{m=1}^{M_k} x_n^{k,m} = \sum_{k=1}^{K}\sum_{m=1}^{M_k} a_n^{k,m}\cos\varphi_n^{k,m} \tag{1. 7a}$$

where x^k now stands for the k^{th} harmonic sinusoid, M_k the number of its partials, and $x^{k,m}$ its m^{th} partial. In the end, we allow a *residue* term r as a component that makes up the difference between the harmonic sinusoids and the real-world audio:

$$x_n = \sum_{k=1}^{K}\sum_{m=1}^{M_k} a_n^{k,m}\cos\varphi_n^{k,m} + r_n \tag{1. 7b}$$

The residue r represents the combination of non-harmonic sinusoids, transients, and noise. Equations (1.7a) and (1.7b) complete the definition of harmonic sinusoid model.

In Fourier-transform-based harmonic sinusoid analysis, it is convenient to use the complex exponential form of sinusoids:

$$\cos\varphi \quad \Leftrightarrow \quad \frac{e^{j\varphi}+e^{-j\varphi}}{2} \tag{1. 8}$$

so that the m^{th} partial of (1.3) becomes

$$x_n^m = \frac{a_n^m}{2}e^{j\varphi_n^m} + \frac{a_n^m}{2}e^{-j\varphi_n^m} \tag{1. 9}$$

That is, a real partial appears as two symmetric complex partials, whose frequencies are symmetric regarding 0.

Throughout this thesis we use the singular form “harmonic sinusoid” to refer to the signal defined in (1.1a) or (1.3), although it contains multiple sinusoids that are harmonically related. The plural form “harmonic sinusoids” is used for multiple signals, each of which is defined in (1.1a) or (1.3).

1.2 Examples of harmonic sinusoids

In this section we give examples of harmonic sinusoids, synthesized by summing up sinusoid partials, each of which is a time-varying sinusoid. A time-varying sinusoid x is synthesized by first synthesizing the amplitude a_n and phase angle φ_n, then calculating $x_n=a_n\cos\varphi_n$. In natural musical sounds there is often a coupling between the amplitude and frequency laws. However, in this section we first study the laws independently, and leave the discussion on their combination to 1.2.3.

In this thesis all raw waveform audio materials are mono, sampled at 44.1kHz, and quantized using 16 bits to fit into the interval -32768~32767. In all waveform figures the horizontal axes indicate time, and the vertical axes indicate displacement (air pressure, or its derivative, in acoustics). In all spectrogram figures the horizontal axes indicate time, and the vertical axes indicate frequency.

1.2.1 Amplitude laws

Example A1: Linear amplitude

A linear amplitude is expressed as

$$a_n = a_0 + A_1 n,\ n\text{=0, 1, …, }N \qquad (1.\ 10)$$

Linear amplitudes are used in McAulay-Quatieri synthesis for interpolating between two amplitude estimates. Real signals rarely follow this model. In machine-driven instruments, such as the pipe organ, it is possible to generate sinusoidal components with highly stable amplitudes, each of which has an approximately constant amplitude, i.e. A_1=0. Figures 1.1 (a) and (b) show the waveform and spectrogram of a constant sinusoid.

Example A2: exponentially decaying amplitude

An exponentially decaying sinusoid has one parameter τ>0 indicating the decay rate, known as the *time constant*:

$$a_n = a_0 e^{-n/\tau},\ n\text{=0, 1, …, }N \qquad (1.\ 11)$$

A small τ indicates a fast decay. When τ→∞ it becomes a constant amplitude. Exponentially decaying sinusoid is found in free vibrating ideal strings. Figures 1.1 (c) and (d) show the waveform and spectrogram of an exponentially decaying sinusoid of length 1s and time constant 0.166s.

Example A3: sinusoid modulated amplitude

Amplitude modulation (AM) is done by multiplying a sinusoid (carrier) with a slow-varying function (modulator). The exponential amplitude, for instance, can be regarded as using an exponential function as modulator. In musical audio a note amplitude-modulated by a periodical modulator is known as a *tremolo*. In this example the modulator is a sinusoid added to a DC shift:

$$a_n = a(1 + d_{AM}\cos(2\pi f_{AM} n + \varphi_{AM})),\ 0 < d_{AM} < 1,\ n\text{=0, 1, …, }N \qquad (1.\ 12)$$

d_{AM}, f_{AM} and φ_{AM} are the amplitude, frequency and starting phase of the modulator. d_{AM} is known as the *modulating depth*. A depth of 0 implies no modulation. Figures

1.1 (e) and (f) show the waveform and spectrogram of a sinusoid-amplitude-modulated sinusoid of length 1s, with modulating depth 0.6 and modulator period 0.2s.

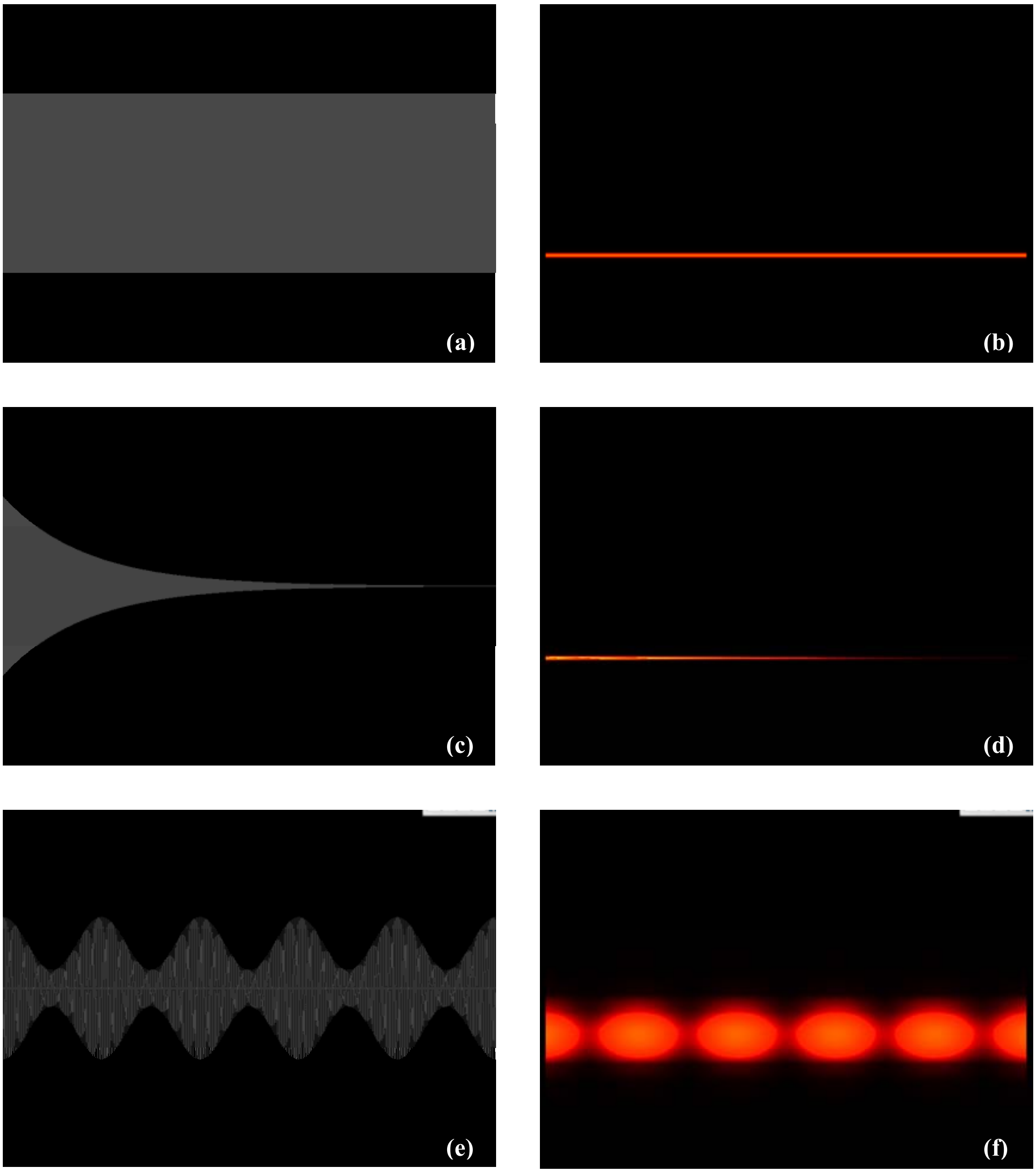


Figure 1. 1 Example amplitude laws

(a)(b) constant; (c)(d) exponential; (e)(f) sinusoid-modulated

Two or more very close sinusoids may also appear as a single sinusoid with amplitude modulation, known as *beats*. Indeed, any signal in the form of (1.12) can be written as the combination of three sinusoids evenly separated in frequency by f_{AM}.

1.2.2 Example frequency laws

Example F1: constant frequency

All the three examples given in 1.2.1 have constant frequencies. A constant frequency appear in the spectrogram as a narrow band centred at a straight line parallel to the time axis, or normal to the frequency axis.

Constant frequencies are very common in music. Free-vibrating instruments, e.g. the piano, guitar, most mallet percussions, are constant-frequency by design (although it's possible to alter the frequency through performing techniques). A big family of wind instruments feature the pitch control using keys/holes. Even for those instruments that feature continuous pitch variation, such as the strings and the trombone, it is rarely hard to generate a stable pitch. However, constant frequency in human voice requires some training.

The phase angle of a constant-frequency sinusoid is a linear function of time.

Example F2: linear and quadratic chirps

The *linear chirp* has a linear frequency and quadratic phase:

$$f(t)=f(0)+bt\,,\ \varphi_n=\varphi_0+2\pi\left(f(0)n+\frac{bn^2}{2}\right)\,,\ n=0,\ 1,\ \ldots,\ N \qquad (1.\ 13)$$

The *quadratic chirp* has a quadratic frequency and cubic phase:

$$f(t)=f(0)+bt+ct^2\,,\ \varphi_n=\varphi_0+2\pi\left(f(0)n+\frac{bn^2}{2}+\frac{cn^2}{3}\right),\ n=0,\ 1,\ \ldots,\ N \quad (1.\ 14)$$

Natural sounds rarely have linear or quadratic frequencies. In McAulay-Quatieri synthesis, they have been used for interpolating between frequency estimates. The linear frequency is used for resynthesis without phase [MQ84]; the quadratic frequency is used for resynthesis with phase [MQ86].

The *linear chirp* has been widely studied in the attempt to model time-varying sinusoids, as it is the simplest case of frequency variation, with all derivatives above the 1st-order being zero. The instantaneous frequency of a linear chirp can be accurately measured using the reassignment [AF95] or least-square-error (§3.1) methods. Figures 1.2 (a) and (b) show the waveform and spectrogram of a linear chirp. The increasing density of the waveform implies a decreasing period. The linear variation of frequency is clearly seen from the spectrogram. Figures 1.2 (c) and (d) show the waveform and spectrogram of a quadratic chirp.

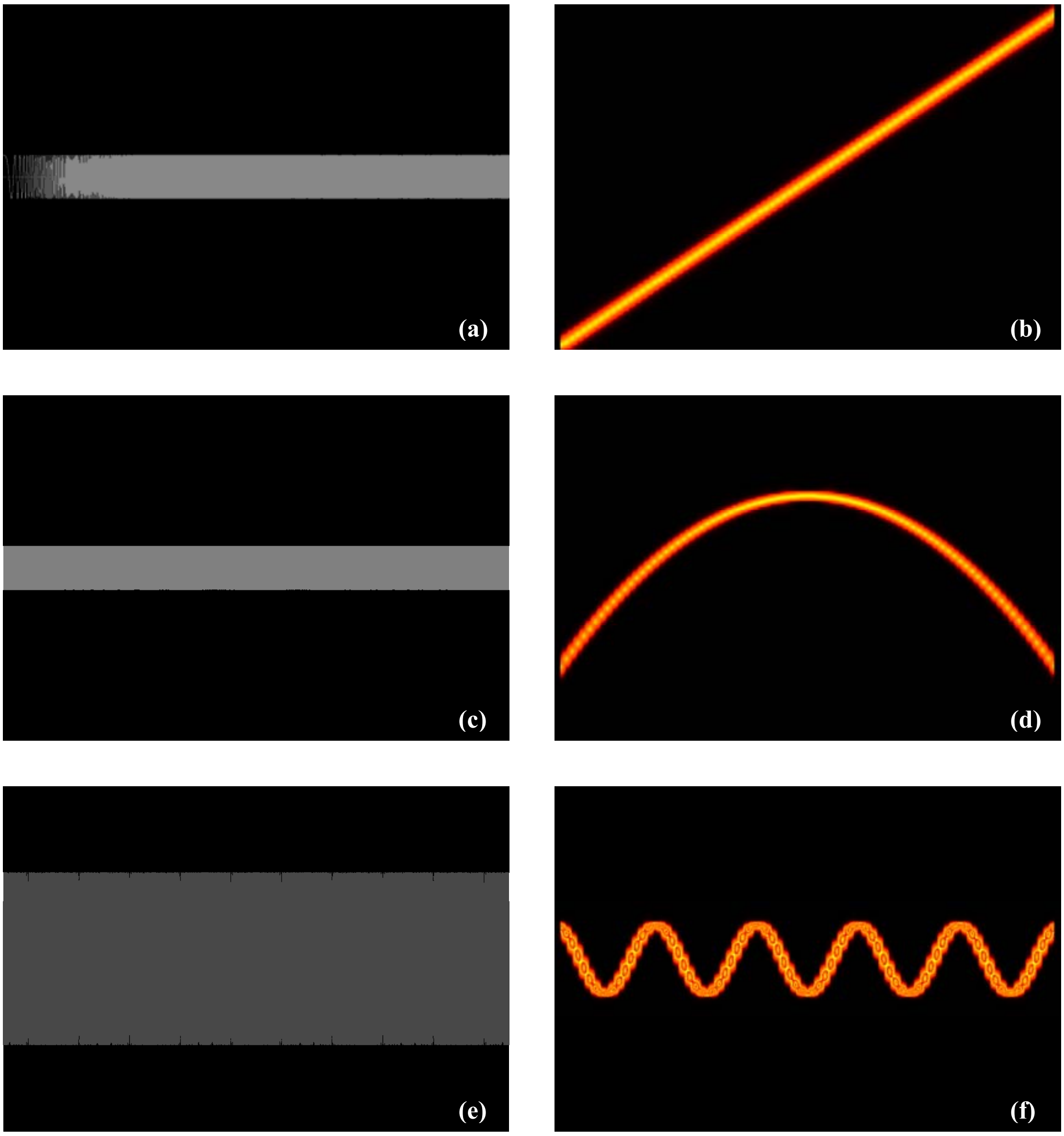


Figure 1. 2 Frequency variation laws

(a) (b) linear; (c)(d) quadratic; (e)(f) sinusoid-modulated

Example F3: sinusoid modulated frequency

Frequency modulation (FM) is done by varying the instantaneous frequency of a sinusoid (carrier) by a time-dependent amount (modulator). FM is well known in radio broadcasting and sound synthesis. In music audio a periodical frequency modulation is known as a *vibrato*. Vibrato is a standard performing technique for bowed string instruments, some wind instruments, as well as for singing. In this example the modulator is a sinusoid:

$$f(t) = f_{carr}(1 + a_{FM}\cos(2\pi f_{FM}t + \varphi_{FM}))$$

$$\varphi_n = \varphi_0 - \frac{a_{FM}f_{carr}}{f_{FM}}\sin\varphi_{FM} + 2\pi f_{carr}n + \frac{a_{FM}f_{carr}}{f_{FM}}\sin(2\pi f_{FM}n + \varphi_{FM}), \qquad (1.\ 15)$$

$$n = 0, 1, \cdots, N$$

where f_{carr} is the carrier frequency, a_{FM}, f_{FM} and φ_{FM} are the amplitude, frequency and starting phase of the modulator. In real music signals a_{FM}<<1, f_{FM}<<f_{carr}. Figures 1.2 (e) and (f) show the waveform and spectrogram of a sinusoid-frequency-modulated sinusoid of length 1s, with modulator period 0.2s. The spectrogram clearly reveals the ongoing process being a frequency modulated sinusoid.

1.2.3 Example harmonic sinusoids

According to (1.1a), a harmonic sinusoid can be synthesized by summing up a individual sinusoids with harmonic frequencies. However, the harmonicity alone does not fully characterise the coupling between partials of real-world music sounds. On one hand, there is inharmonicity which shifts partial frequencies from perfect harmonicity; on the other hand, there is also the coupling between partial amplitudes, and the coupling between amplitude and frequency, to be considered.

Example 1: simple harmonic sinusoids (formants)

We call the wave function of simple harmonic vibration a *simple harmonic sinusoid*. This family of waveforms are also found in other vibration types, such as in a periodically stimulated resonator. All partials of simple harmonic sinusoids have constant amplitudes and frequencies, and all partial frequencies are in perfect harmonicity. The stable sound of a pipe organ is very close to a simple harmonic sinusoid. A simple harmonic sinusoids is written as

$$x_n = \sum_{m=1}^{M} a^m \cos(2\pi m f^1 n + \varphi^m)\,,\ n\text{=0, 1, …, } N \qquad (1.\ 16a)$$

where M is the number of partials.

(1.16a) has $2M$+2 parameters, i.e. M, M partial amplitudes a^m (m=1, …, M), M partial phases φ^m (m=1, …, M), and the fundamental frequency f^1. Most acoustic instruments show a decreasing trend of partial amplitude regarding partial index, while local variations can be quite unpredictable. As an example we initialize the partial amplitudes as

$$a^m = A\frac{|\text{sinc}\, m\theta|}{m},\quad m = 1, 2, \cdots, M \qquad (1.\ 16b)$$

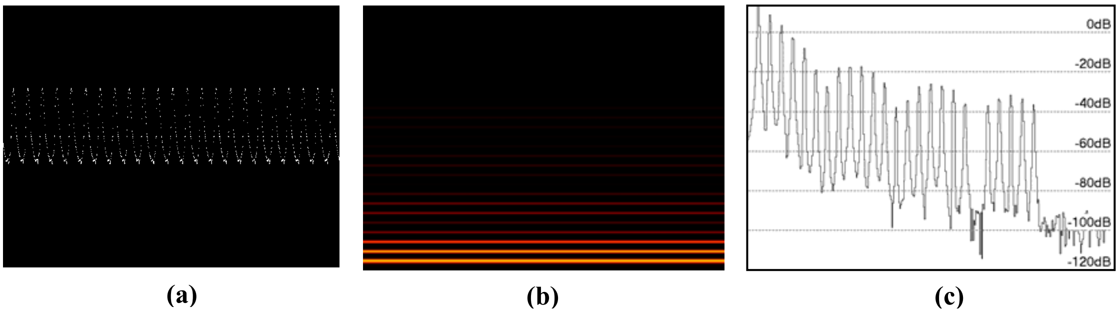


Figure 1. 3 Simple harmonic sinusoids
(a) waveform; (b) spectrogram; (c) spectrum

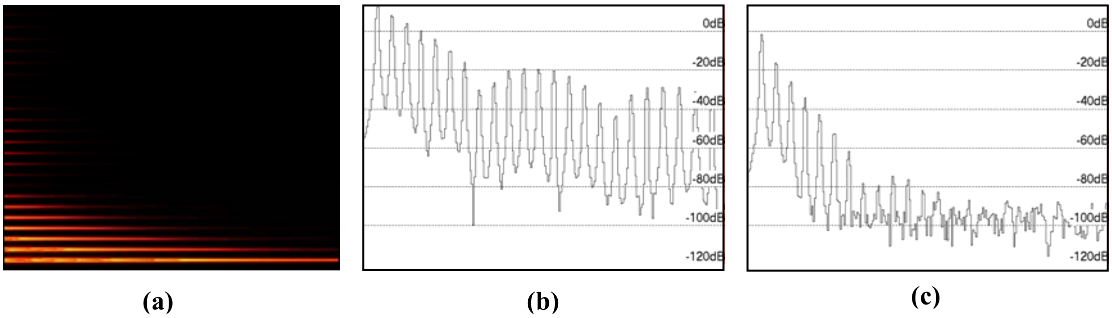
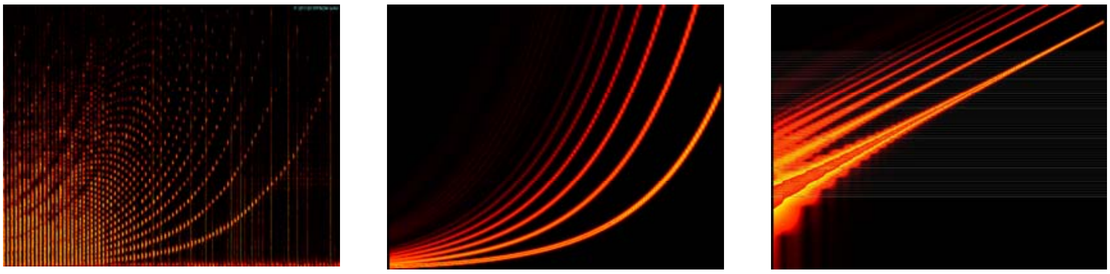
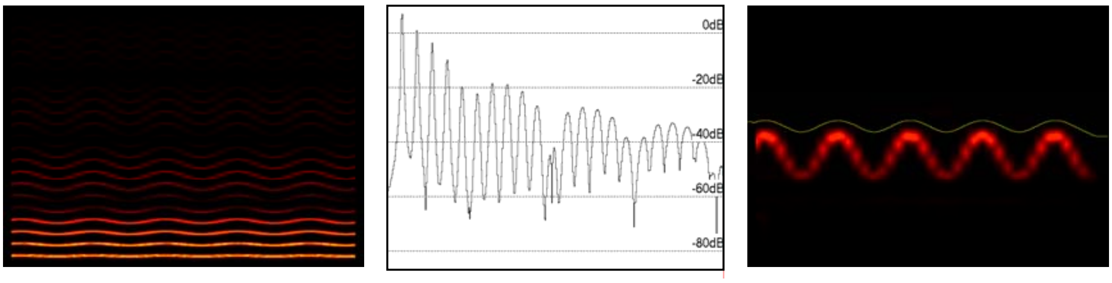


Figure 1. 4 A4 Harmonic sinusoids with exponential decay
(a) spectrogram; (b)(c) spectra at time 0 and 2s



(a) **(b)** **(c)**

Figure 1. 5 Exponential chirp
(a) chromatic piano scale; (b)(c) exponential chirps on linear and logarithmic frequency axes



(a) **(b)** **(c)**

Figure 1. 6 Vibrato accompanied by tremolo
(a) spectrogram; (b) spectrum at 0.2s; (c) the 6th partial

where A>0, 0<θ<0.5 and sinc is the *continuous sinc function*, defined as (A.4a). The physical model behind (1.16b) is an ideal string with fixed ends, stimulated at position θ from one end.

(1.16b) shows two typical phenomena of pitched events: the decay of amplitude with partial index, and the resonance at specific frequency ranges, known as *formants*. In (1.16b) the formants are found where $m\theta$ is close to 0.5, 1.5, 2.5, etc. In particular, when θ=0.5, the even partials disappear and the odd partials decay like m^{-2}.

Figure 1.3 depicts the waveform, spectrogram and spectrum of a simple harmonic sinusoid with M=25, θ=0.15. The formant structure is clearly seem in the spectrum. The 20th partial disappears since 20θ is an integer.

Example 2: Piano-like harmonic sinusoids (partial-dependent decay)

A simple harmonic sinusoid is made piano-like by modifying the partials so that they

1) have constant frequencies that conforms to a stiff string model;

2) have decaying amplitudes so that lower partials generally decay slower than higher ones.

[FR98] gives the stiff string frequency model as

$$f^{m} = mf^{1}\sqrt{1+B(m^{2}-1)} \qquad (1.\ 17a)$$

where B is a stiffness coefficient, typically $0 \le B < 0.001$. It is observed that inharmonicity grows larger as the string becomes shorter (and pitch becomes higher, in general).

In actual piano sound the decay of individual partials can be highly complicated, partially due to the presence of multiple strings at unison. As a result only some low partials roughly show exponential law. In this simple example we use the exponential law on all partials. [FR98] shows that the time constant of the fundamental partial drops roughly like $F0^{-1}$, and the time constant of partials of the same note drops roughly like $(f^{m})^{-0.5}$. Based on these rules, the time constant of the mth partial of pitch $F0$ is

$$\tau_{F0}^{m} = \tau_{A4}^{1} \cdot (\mathrm{F0/440Hz})^{-1} \cdot \mathrm{m}^{-1/2} \cdot (1+B(\mathrm{m}^{2}-1))^{-1/4}. \qquad (1.\ 17b)$$

where τ_{A4}^{1} the time constant of the fundamental partial of central A (i.e. A4=440Hz). Finally we have the partial-dependent amplitude rule

$$a_{n}^{m} = A\frac{\left|\operatorname{sinc} m\theta\right|}{m}e^{-\frac{(F0/440Hz)m^{1/2}(1+B(m^{2}-1))^{1/4}}{\tau_{A4}^{1}}n}. \qquad (1.\ 17c)$$

Figure 1.4 shows the spectrogram, spectrum at the beginning, and spectrum after 2 seconds, of a harmonic sinusoid synthesized using M=25, $F0$=440Hz, θ=0.12, and τ_{A4}^{1}=1s. The varying shape of the spectrum implies a change of *short-term timbre* over time.

Example 3: exponential chirp (glissando)

The exponential chirp, in which the pitch varies as an exponential function of time, is known as a *glissando* in music. Figure 1.5 (a) shows the spectrogram of a chromatic piano scale (A0~C8), played at roughly constant intervals, which can be regarded as a simulation of true glissando.

In this example we initialize the partial amplitudes using (1.24b) with θ=0.12 and let them be constant, and let partial frequencies be perfectly harmonic. The exponential frequency is written as

$$f^m(t) = mf^1(0)\cdot 2^{t/\tau_2},\ \varphi_n^m = \varphi_0^m + \frac{2}{\ln 2}\pi mf^1(0)\tau_2\cdot(2^{n/\tau_2}-1) \qquad (1.\ 18)$$

where τ_2 is the duration in which the pitch progresses by one octave. Figures 1.5 (b) and (c) show the spectrograms, in linear and logarithmic frequency scales respectively, of an exponential chirp that progresses from A1 to A7 in 3 seconds

Example 4: vibrato (AM that accompanies FM)

The *vibrato* is frequently encountered in music performance, especially in human voice and bowed string instruments. During a vibrato the pitch repeatedly rise and drop about a central frequency. In real-world signals the vibrato is usually accompanied by a modulation of amplitude, or *tremolo*, which has the same modulation period as the vibrato itself.

This tremolo can be regarded as the joint effect of two mechanisms: a "true" tremolo due to the same performance mechanism as the vibrato, and a "vibrato-bound" tremolo due to filtering effect. In this example we implement the accompanying tremolo using a *source-filter model*, where the source provides a time-dependent factor and the filter provides a frequency-dependent factor:

$$a_n^m = A_m A_s(n) A_f(f^m(n)) \qquad (1.\ 19a)$$

The source factor $A_s(n)$ contributes to an overall amplitude modulation common to all partials (true tremolo), and the filter factor $A_f(f^m(n))$ contributes to a partial-dependent modulation. It is apparent that if f^m is periodical, then $A_f(f^m(n))$ must also be periodical, and has the same period. In the frequency band where $A_f(f)$ is increasing, the amplitude becomes larger as the frequency goes up, and vice versa. A_m is a normalization factor so that $a_0^m = A_m A_s(0) A_f(f^m(0))$, m=1, …, M.

In this example partial frequencies are in perfect harmonicity, and the frequency modulator is a pure sinusoid. The frequency law is

$$f^m(t) = f^m(0)\cdot(1 + a_{FM}\sin 2\pi f_{FM}t),$$

$$\varphi_m^n = \varphi_m^0 + 2\pi f^m(0)n - \frac{a_{FM}f^m(0)}{f_{FM}}(\cos 2\pi f_{FM}n - 1) \tag{1. 19b}$$

We initialize the partial amplitudes $a^m(0)$ using (1.16b) with some $\theta(0)$. Let $A_s(n)$=1. We derive the filter factor $A_f(f)$ by rewriting (1.16b) as

$$a_n^m = A\frac{\left|\operatorname{sinc}\frac{f^m(n)}{f^1(n)}\theta(n)\right|}{m} = \frac{A}{m}\left|\operatorname{sinc} f^m(n)\theta_1\right| \tag{1. 19c}$$

where θ_1=$\theta(n)/f^1(n)$, and let

$$A_m = \frac{A}{m},\ A_f(f) = \left|\operatorname{sinc} f\theta_1\right| \tag{1. 19d}$$

The physical model behind (1.19d) is a vibrato on a bowed string created by varying the string length at the far end from a fixed bowing point. In this case both the frequencies and θ are reciprocal to the string length, i.e. f^m is proportional to θ, therefore θ_1 is a constant, and can be calculated using θ_1=$\theta(0)/f^1(0)$. From a spectral point of view, (1.19d) implies that the positions of formants remain stable.

Figure 1.6 (a) shows the spectrogram of a synthesized vibrato of length 1s, with M=25, $\theta(0)$=0.18, $f^1(0)$=440Hz, a_{FM}=1/22, f_{FM}=5Hz. Figure 1.6 (b) shows its short-time spectrum at 0.2s. In figure 1.6 (c) we draw the 6th partial, together with its amplitude curve in time domain. There is a positive correlation between the amplitude and frequency, which is a result of the sinc function being increasing near 6×0.18=1.08.

1.3 The Fourier transform

The importance of the complex representation of sinusoids lies in the fact that time-invariant complex sinusoids are eigenvectors of linear time-invariant systems, with the frequency response being the eigenvalues. In discrete-time processing, this is represented by the *discrete-time Fourier transform*, or DTFT:

$$X(f)=\sum_n x_n e^{-j2\pi fn}\ ,\ x_n=\int_{-0.5}^{0.5} X(f)e^{j2\pi fn}df \tag{1. 20}$$

f is known as the *digital frequency*, dimensionless, with *f*=1 corresponding to the sampling frequency. The Nyquist frequency interval for *f* is (-0.5, 0.5). In numerical practices we use the *discrete Fourier transform*, or DFT:

$$X_k=\sum_{n=0}^{N-1} x_n e^{-j2\pi kn/N}\ ,\ x_n=\frac{1}{N}\sum_{k=0}^{N-1} X_k e^{j2\pi kn/N}\ . \tag{1. 21}$$

where *N* is the DFT size and *k* is the discrete Fourier frequency in *bin*s, with *k*=*N* bins corresponding to the sampling frequency. X_k is a sampled version of *X*(*f*). Both *X*(*f*) and X_k are called *spectrum*. (1.21) shows that these *N* complex sinusoids, multiplied by $N^{1/2}$, compose an orthonormal basis of the *N*-dimensional Hilbert space. In practice we only consider $N=2^L$, $L\in Z^+$.

1.3.1 Constant sinusoids

Let *x* be a time-invariant discrete complex sinusoid, i.e. $x_n=ae^{j(2\pi fn+\varphi)}$, then its DTFT is

$$X(g)=ae^{j\varphi}\sum_m \delta(g-f-m) \tag{1. 22a}$$

where *δ* is the *Dirac delta function*, defined in Appendix A.1. Within the Nyquist frequency interval (-0.5, 0.5), *X*(*g*) has only one spike, which is located at *f*.

The DFT of *x* calculated on the interval 0≤*n*<*N* is

$$X_k=e^{j(\varphi+\pi(f-k/N)(N-1))}\cdot aN\,\mathrm{sinc}_N(k-Nf) \tag{1. 22b}$$

where $\mathrm{sinc}_N(f)$ is the *N*-point *discrete sinc function*, defined in Appendix A.1. (1.22b) can be interpreted as the frequency-domain convolution of the rectangular window spectrum (i.e. the sinc function) with the spectrum of *x* (1.22a), sampled at *k*/*N*. Since the sinc function is concentrated around 0, X_k is concentrated around the frequency *Nf*, in bins.

1.3.2 Windowed DFT

Let x be defined as above and w be a discrete window function supported on $0 \leq n < N$, with spectrum $W(f)$, then the *windowed DFT* of x using window w is

$$X_k = \sum_{n=0}^{N-1} x_n w_n e^{-j2\pi kn/N} = ae^{j\varphi} W(k/N - f) \tag{1. 23}$$

(1.23) is the starting point for DFT-based parameter estimators. All window functions used in this thesis are real, symmetric, non-negative and low-pass. Some of these are listed in Appendix A.2.

1.3.3 Slow-varying sinusoids

Now we look at the DFT of a slow-varying sinusoid $x_n = a_n e^{j\varphi_n}$. With the following proposition, we show that if the amplitude and frequency variations are slow enough, then its spectrum approximates that of a constant sinusoid at its central amplitude and central frequency.

Proposition 1.1 Let x be a slow-varying sinusoid, i.e. $x_n = a_n e^{j\varphi_n}$, then its windowed DFT can be written as

$$X_k = a_{N/2} e^{j\varphi_{N/2}} W(k/N - f_{N/2}) + \varepsilon_k, \tag{1. 24a}$$

where the term ε_k is bounded by

$$|\varepsilon_k| \leq \sup|f'| \cdot \pi a_{N/2} \sum_{n=0}^{N-1} |w_n| (n - N/2)^2 + \sup|\Delta a| \cdot \sum_{n=0}^{N-1} |w_n (n - N/2)|. \tag{1. 24b}$$

Δa is the difference function of a, i.e. $\Delta a_n = a_n - a_{n-1}$.

The proof, following Mallat's proof for the continuous FT [Mallat99], is given in Appendix A.3.1. The first term on the right side bounds the spectral departure due to frequency variation, while the second term bounds that due to amplitude variation. Factors $\sum_{n=0}^{N-1} |w_n| (n - N/2)^2$ and $\sum_{n=0}^{N-1} |w_n (n - N/2)|$ only depend on the window function. $(n-N/2)^2$ in the first term and $(n-N/2)$ in the second term are both weighed by

$|w_n|$, implying that the approximation of X with $\tilde{X}$ is improved by using a window function that vanishes at both ends.

As a numerical example, we consider a linear amplitude a combined with a linear frequency f, and let a have a total variation of $\mathrm{A}\cdot a_{N/2}$, and f have a total variation of B bins, during the analysis window. Therefore $\sup|f'|=\mathrm{B}/N^2$, $\sup|\Delta a|=\mathrm{A}\cdot a_{N/2}/N$. When a triangular window is used for DFT, we have

$$\sum_{n=0}^{N-1}|w_n| = N/2\,, \tag{1. 25a}$$

$$\sum_{n=0}^{N-1}|w_n(n-N/2)| = \frac{(N+2)(N-2)}{24}\,, \tag{1. 25b}$$

and

$$\sum_{n=0}^{N-1}|w_n|(n-N/2)^2 = \frac{N(N+2)(N-2)}{96}\,. \tag{1. 25c}$$

It follows from (1.24b) that

$$\left|X_k-\tilde{X}_k\right| \le a_{N/2}\pi\frac{\mathrm{B}}{N^2}\frac{N(N+2)(N-2)}{96}+a_{N/2}\frac{\mathrm{A}}{N}\frac{(N+2)(N-2)}{24} < \frac{a_{N/2}N}{2}\frac{\pi\mathrm{B}+4\mathrm{A}}{48}\,. \tag{1. 25d}$$

$\dfrac{\pi\mathrm{B}+4\mathrm{A}}{48}$ is interpreted as the relative error bound, since $\dfrac{a_{N/2}N}{2}$ is the amplitude at the main spectral peak.

Proposition 1.1 also provides a criterion for comparing amplitude variation with frequency variation in their effects on the spectrum. For example, (1.25d) indicates that 1/π bins frequency variation is “equally bad” as 25% amplitude variation in their contributions to the error bound.

It is apparent in (1.25d) that the error bound grows very large when B is more than 2 or 3 bins. $\tilde{X}$ becomes a poor approximation of X when the frequency variation is on the magnitude of bins, because different parts of x are now contributing to different bins. This can be interpreted either as that x varies too fast for the high frequency resolution of the window, or as that the window is too long to capture the dynamics of

x. Accordingly, to better represent x, a shorter window with a lower frequency resolution shall be used. For the linear amplitude and linear frequency example, let B=3 and A=0.5, then the relative error bound in (1.25d) is 23.8%. If the window is shortened by half, then A is reduced to 0.286 for the first half and 0.222 for the second half, B is reduce to 0.75, so that the relative error bound is reduced to 7.3% for the first half and 6.8% for the second half.

The following proposition is an example of dividing a long window into shorter ones to study the spectral properties of a sinusoid with fast frequency variation. It shows that the spectrum of a time-varying sinusoid does not spread far from its instantaneous frequencies during the analysis window on which the spectrum is calculated.

Proposition 1.2 Let x be a sinusoid with constant amplitude and varying frequency, i.e. $x_n = e^{j\varphi_n}$, its instantaneous frequency f be within the interval F=(f_1, f_2) during [0, N], and X be its DFT. Given an integer L, 3≤L<<N, the amplitude spectrum at k is bounded by

$$|X_k| < (0.01 + 0.112 / L)N + \frac{\pi^3 N^3}{60L^2} \sup|f'| \tag{1. 26}$$

if k is at least 1.5L bins from F, i.e. $0 \le k < Nf_1 - 1.5L$ or $Nf_2 + 1.5L < k < N/2 - 1$.

This is proved by breaking the window of size N into L-1 overlapping windows of size 2N/L. The complete proof is given in Appendix A.3.2. The right hand side is an upper bound on the spectral leakage outside the instantaneous frequency range. Of the three terms in (1.16), the last term shows how the decrease in window size helps suppressing the leakage due to frequency variation. The first and seconds terms represent the “original” leakage of a stationary sinusoid, with the second term mostly contributed by the use of rectangular windowing for calculating X. By using fade-in and fade-out at both ends, we get the following result with a smaller bound. The proof is discussed in Appendix A.3.2.

Corollary 1.3 Let x be a sinusoid with constant amplitude and varying frequency, i.e. $x_n = e^{j\varphi_n}$, its instantaneous frequency f be within the interval F=[f_1, f_2] during [0, N]. Given an integer L, 3≤L<<N, define a window function

$$w_n = \begin{cases} \sin^2 \dfrac{\pi n}{2M}, & 0 \le n < M, \\ \sin^2 \dfrac{\pi(n-N)}{2M}, & N - M \le n < N \\ 1, & otherwise \end{cases}, \quad (1.\ 27a)$$

where M=N/L, and let X be the DFT of x calculated using window w, then

$$|X_k| < 0.01N + \frac{\pi^3 N^3}{60L^2} \sup|f'| \quad (1.\ 27b)$$

if k is at least 1.5M bins from the F, i.e. $0 \le k < Nf_1 - 1.5M$ or $Nf_2 + 1.5M < k < N/2 - 1$.

Regarding spectral energy, Carson's bandwidth rule [Carson22, Carlson81] of FM communication provides a practical approximation. Carson's rule states that more than 98% the total energy of a frequency-modulated signal is distributed within Carson's FM bandwidth, given as $CRB = 2(f_\Delta + f_M)$, where f_Δ is the maximal deviation of the instantaneous frequency from some central frequency, and f_M is the maximal frequency of the modulator.

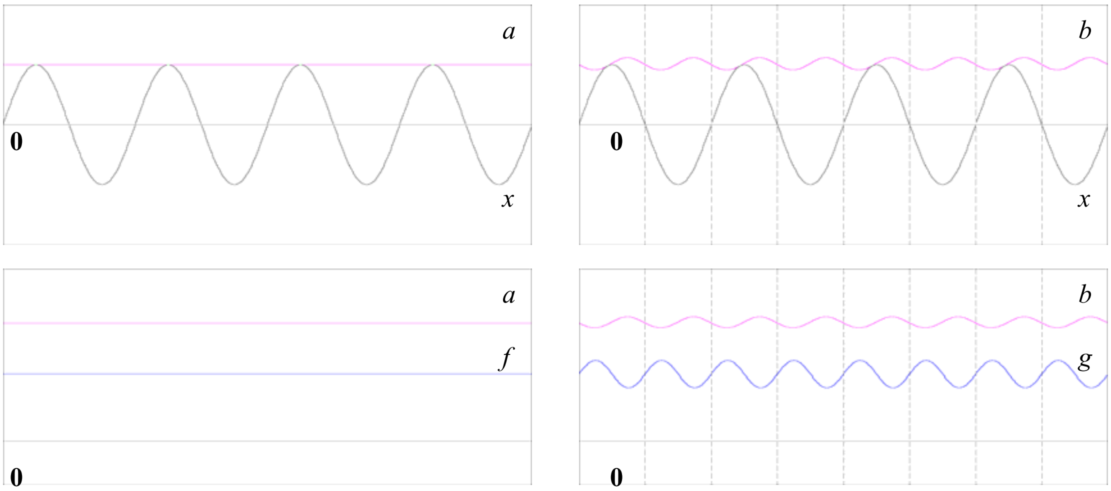
1.4 Uniqueness and slowness

From (1.7b), it is obvious that the analysis problem is underdetermined, i.e. the model contains more data than the modeled signal. Therefore there is no unique solution for the analyzer. This is true even when we model a single sinusoid (M=1). Suppose x is a time-varying sinusoid with amplitude $a(t)$>0 and phase angle $\varphi(t)$, i.e. $x(t)=a(t)\cos\varphi(t)$. Let t_1 and t_2 be two adjacent zeros of x, so that x remain positive within (t_1, t_2). Let $\varphi(t_1)=-\pi/2$, $\varphi(t_2)=\pi/2$. Now let $\theta(t)$ be a continuous differentiable monotonic function and $\theta(t_1)=-\pi/2$, $\theta(t_2)=\pi/2$, and let b be defined on $[t_0, t_1]$ and

$$b(t) = \begin{cases} a(t)\dfrac{\varphi'(t)}{\theta'(t)}, & t = t_0 \ or \ t = t_1 \\ a(t)\dfrac{\cos\varphi(t)}{\cos\theta(t)}, & otherwise \end{cases} \quad (1.\ 28)$$

then $x(t) = b(t)\cos\theta(t)$, i.e. b and θ are also the instantaneous amplitude and phase angle of x. This implies that the instantaneous frequency is unique only when the instantaneous amplitude is determined, and vice versa. Given a signal $x(t)$, we can always trade amplitude for frequency, provided that continuity is preserved at zeros of x.

This non-uniqueness raises a question on the ground truths for sinusoid modeling. Usually when a time-varying sinusoid is synthesized from artificial amplitude and frequency laws and used in tests, the designed amplitude, frequency and phase are regarded as the ground truth, to which the parameter estimates are compared. However, since the sinusoid representation is not unique, there exist other combinations of parameters which are no less "true" than the designed ones. Accordingly the designed parameters alone are not enough to serve as the ground truth. In other words, a parameter set being closer to the designed one does not guarantee its being more accurate. This difficulty is partially relieved by knowing that the parameters are *slow-varying*. For example, Figure 1.7 shows a constant sinusoid with no amplitude or frequency variation. In (a) the instantaneous amplitude a is compared with the sinusoid x in the upper graph, and with the instantaneous frequency f in the lower graph. In (b) we show another choice of the amplitude b and frequency g, which produces the same signal x. Under the slow-varying assumption, (a) is considered a better representation since it shows slower parameter variation. A parameter estimator that assumes slow-variation of sinusoids is more likely to produce the amplitude and frequency lines in (a).



(a) (b)

Figure 1. 7 A constant sinusoid

(a) constant-parameter representation; (b) non-constant-parameter representation

The following proposition shows that if the variation of amplitude and frequency are very slow, then *a* approximate *b*, *f* approximates *g*.

Proposition 1.4 Let $x=a\cos\varphi=b\cos\theta$, $0\le\frac{|a'|}{a},\frac{|b'|}{b}\le A_1$, $0\le\frac{|a''|}{a},\frac{|b''|}{b}\le A_2$, $\varphi'=2\pi f>0$, $\theta'=2\pi g>0$, $0\le|f'|,|g'|\le F_1$, $\varphi(-\tau)=\theta(-\tau)=-\frac{\pi}{2}$, $\varphi(\tau)=\theta(\tau)=\frac{\pi}{2}$, then

$$|f-g|\le\min\left(\frac{A_2+2\pi(g\tan|\theta|+f\tan|\varphi|)A_1+\pi(\tan|\theta|+\tan|\varphi|)F_1}{2\pi^2(f+g)},2\tau F_1\right),\tag{1. 29a}$$

$$|\varphi-\theta|\le 2\pi\tau^2F_1,\tag{1. 29b}$$

$$\frac{|a-b|}{b}\le\frac{\sin\max(|\theta|,|\varphi|)}{\cos\varphi}2\pi\tau^2F_1,\text{ when }x\neq 0.\tag{1. 29c}$$

The proof is given in Appendix A.3.3. Proposition 1.4 implies that if the variations of both {*a*, *f*} and {*b*, *g*} are very slow, then they are similar in value. An exception is the amplitude at zeros of *x*, where the relative error between *a* and *b* is not bounded by (1.29c). However, it is easy to derive from (1.28) that at zeros of *x* the following holds

$$\frac{|a-b|}{b}=\frac{|g-f|}{f}\le\frac{2\tau F_1}{f}, \quad (1.\ 29d)$$

so that the error between *a* and *b* is still bounded.

In sinusoid modeling there has not been much discussion on how the *slowness* of parameters should be measured or how parameter estimation is affected by it. Intuitively, the derivatives of amplitude and frequency measure the speed of their variations. We therefore compose the following function to evaluate how fast the variations are:

$$I=\int(\eta a'(t)^2+(1-\eta)\varphi''(t)^2)dt \quad (1.\ 30a)$$

where a' and $\varphi''=2\pi f'$ respectively measures the amplitude and frequency variation rates, and $\eta\in(0,1)$ is a balancing factor that trades amplitude variation for frequency variation. One plausible selection of η is using the spectral error bound (1.24b), so that $\eta^{1/2}a'$ contributes the same amount to the error bound as $(1\text{-}\eta)^{1/2}\varphi''$. Another interpretation of the criterion (1.30a) is the *high-frequency energy*:

$$I=\frac{1}{2\pi}\eta\int\omega^2|A(\omega)|^2d\omega+2\pi(1-\eta)\int\omega^2|F(\omega)|^2d\omega \quad (1.\ 30b)$$

where *A* and *F* are the Fourier transforms of *a* and *f*, respectively.

The necessary condition for *a* and *f* to be “slowest-varying”, i.e. minimizing (1.30a), is

$$-\eta a''a\tan\varphi+(1-\eta)\varphi^{(4)}=0 \quad (1.\ 30c)$$

The proof is given in Appendix A.3.4. The discrete-time version of (1.30a) and (1.30c) is

$$I=\sum_n\eta(\Delta a)^2+(1-\eta)(\Delta^2\varphi)^2, \quad (1.\ 31a)$$

$$-\eta a_n\Delta^2a_n\tan\varphi_n+(1-\eta)\Delta^4\varphi_n=0, \quad (1.\ 31b)$$

where Δ, Δ^2 and Δ^4 are 1st-, 2nd- and 4th-order difference operators. In both (1.30c) and (1.31b) the amplitude and phase angle are closely coupled through the factor $\eta\tan\varphi$. In synthesizing time-varying sinusoids, when amplitude and frequency laws are designed without this coupling, the a and f we create are usually not slowest-varying parameters of x. The only exception is when $a''=\varphi^{(4)}=0$, i.e. the amplitude law is linear *and* frequency law is quadratic. This coincides with the piecewise sinusoids in McAulay-Quatieri synthesis, which uses linear interpolation of amplitudes, and trinomial interpolation of phase angles.

Due to the uniqueness problem, we always avoid evaluating parameter estimators by direct comparison of "true" and estimated parameters. Time domain signals are compared instead.

1.5 Harmonic sinusoid modeling

Harmonic sinusoid modeling is the process of converting between the waveform representation and the harmonic sinusoid representation. The device that converts a signal from waveform representation to harmonic sinusoid representation is a *harmonic sinusoid analyzer*. The device that converts a signal from the harmonic sinusoid representation to waveform representation is a *harmonic sinusoid synthesizer*. The analyzer and the synthesizer make up the complete harmonic sinusoid modeling system (Figure 1.8).

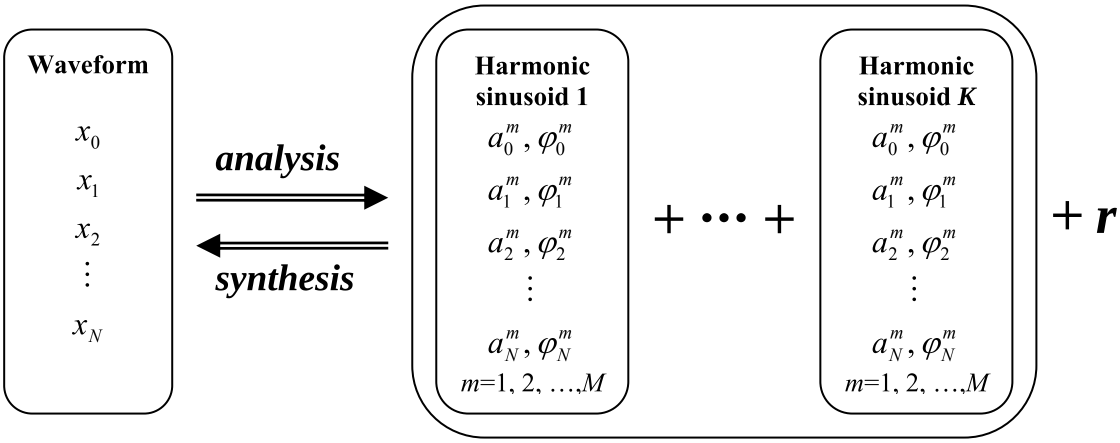


Figure 1. 8 Harmonic sinusoid modeling

As seen in (1.30b), the slow-varying parameters do not contain substantial high-frequency energy. Therefore by doing a subsampling of a, f and φ, we are able to get a more compact representation that preserves most sinusoidal features. In the simplest case the sampling points are uniformly distributed on the time axis. The three parameters are measured at each of these points from an interval centred at it, known as a *frame*. The uniform interval between adjacent frame centres is the *hop size*. In this thesis we fix the hop size at half the frame size, i.e. frames have 50% overlap, as shown in Figure 1.9.

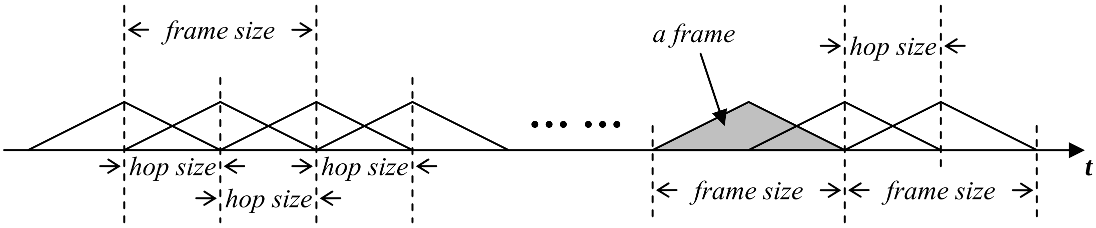


Figure 1. 9 Frames with 50% overlap

The three parameters measured from a frame compose a *short-time sinusoid atom*, or *atom*. Multiple atoms can be estimated from the same frame, distinguished by frequency. Each partial of a harmonic sinusoid appears as one atom at every frame within its duration. All atoms of a harmonic sinusoid at the same frame form a *harmonic sinusoid particle*, or *harmonic particle*. Atoms within a harmonic particle are distinguished and sorted by partial index. Harmonic particles characterize the spectral harmonicity structure of a harmonic sinusoid. All atoms of the same partial form a *sinusoid track*. Atoms within a sinusoid track are distinguished and sorted by *frame index*. Sinusoid tracks characterize the time continuity structure of a harmonic sinusoid. Spectral harmonicity and time continuity are the two main features we explore in harmonic sinusoid modeling.

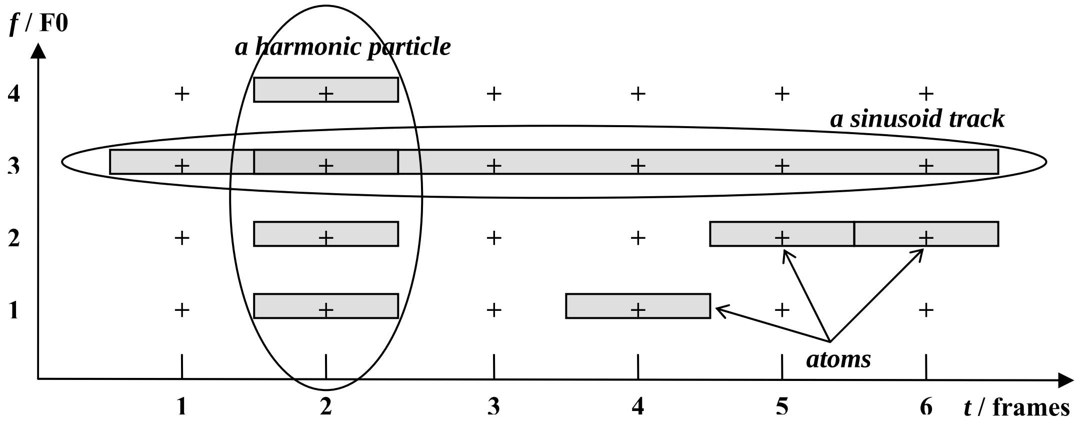


Figure 1. 10 Structure of a harmonic sinusoid

Figure 1.10 depicts the components of a harmonic sinusoid in time-frequency plane. The fundamental partial is given partial index 1. Each atom virtually represents a time-frequency area which is narrow in frequency and wide in time. As a result the sinusoid track covers a narrow band in the plane which is continuous in time, but the harmonic particle consists of only isolated atoms and remains sparse in frequency. The harmonic sinusoid, as a whole, is dense in time and sparse in frequency.

Given an audio signal, the frame-based harmonic sinusoid analyzer extracts atoms and form harmonic sinusoids from them. The harmonic sinusoid can either be regarded as a collection of harmonic particles sorted in time by frame index, or be regarded as a collection of sinusoid tracks sorted in frequency by partial index. In this thesis we use the first interpretation to implement the analyzer. More details will be discussed in Chapter 2 after a brief review of standard sinusoid modeling techniques.

1.6 Summary

In this chapter we have defined the harmonic sinusoid model using time-varying sinusoids, and discussed several aspects of the analysis of time-varying sinusoids. By forcing harmonicity between sinusoids, the harmonic sinusoid representation enables the direct modeling of harmonic sounds, as shown by our examples. Discussions on time-varying sinusoids in this and future chapters will lead to improved sinusoid estimation methods and parameter evaluation designs.

Chapter 2

Review of related techniques

In this chapter we briefly review the techniques related to sinusoid modeling. The standard sinusoid model (plus noise) [MQ86, Serra89] is given as follows.

$$x_n = \sum_{m=1}^{M} x_n^m + r_n = \sum_{m=1}^{M} a_n^m \cos\varphi_n^m + r_n\,,\ \varphi_n^m = \varphi_0^m + 2\pi\int_0^n f^m(t)dt \tag{2.1}$$

x^m is a time-varying sinusoid, known as the m^{th} partial. Compared with (1.7b), the standard sinusoid model constructs a sound directly from individual sinusoids without a mid-level structure of harmonic sinusoids.

The sinusoid modeling system converts a signal between the waveform and sinusoid-plus-noise representations. The analyzer finds sinusoids from the waveform. Inside the analyzer a peak picker finds short-time sinusoid atoms and estimates their parameters, and a peak tracker connects the found atoms into sinusoid tracks. The synthesizer reconstructs a waveform from the sinusoid tracks. Signal contents that are not represented in the sinusoid tracks are left in the residue *r*.

The sinusoid modeling is based on short-time sinusoid atoms. In the presence of concurrent sinusoids and noise, it is desirable to isolate the sinusoid of interest from other events before estimating the parameters. In practice, the DFT, which implements a dense bank of band-pass filters, is used as the starting point of sinusoid analysis.

This chapter is arranged as follows. 2.1 discusses the detection of sinusoid atoms; 2.2 and 2.3 review several methods for measuring sinusoidal parameters; 2.4 reviews the methods for tracking sinusoids over time; 2.5 reviews synthesis techniques. In 2.6 we draw a comparison between standard and harmonic sinusoid models.

2.1 Detecting sinusoids from DFT

As discussed in §1.3, the energy of slow-varying sinusoids is concentrated within a narrow band of several DFT bins, and therefore can be detected as *spectral peaks*. DFT-based sinusoid detector finds sinusoids from the signal by locating these peaks. This section discusses several aspects of this "peak picking" method.

2.1.1 Constant sinusoid with noise

Let X be the DFT of x. A local spectral peak is defined as a bin k where $|X_k|\geq|X_{k+1}|$, $|X_k|\geq|X_{k-1}|$ (but at least one identity does not hold). A global spectral peak is defined as a local spectral peak at bin k so that $|X_k|\geq|X_l|$, $\forall l$. We always expect a sinusoid to appear in the amplitude spectrum as a local peak. This is guaranteed for constant complex sinusoids by the following.

Proposition 2.1 (windowed DFT): If the window spectrum $W(f)$ satisfies that $|W(f)|>|W(g)|$, $\forall |f|\leq 0.5/N$, $0.5/N<|g|<1\text{-}0.5/N$, $N\cdot(f-g)\in\mathrm{Z}$, then a constant complex sinusoid $x_n=ae^{j(2\pi fn+\varphi)}$ $(0\leq f<0.5)$ has a global windowed DFT peak at bin k, where k/N is closest to f.

The proof is given in Appendix B.1.1. All the "usual" window functions we consider have monotonically decreasing amplitude on the interval $[0, 0.5/N]$, and $|W(0.5/N)|$ is larger than $|W(f)|$ for all $0.5/N<f<1\text{-}0.5/N$, so that the requirements on W are fully satisfied. In fact, the condition $N\cdot(f-g)\in\mathrm{Z}$ indicates that we only need to compare values of W sampled at intervals of whole bins, e.g. $W(0.1/N)$ with $W(1.1/N)$, $W(2.1/N)$, etc. There is no need to require $W(0)>W(1.1/N)$. This shows the effect of frequency sampling when we work with the DFT. For usual window functions, the amplitude at 0.5 bin becomes a measure of the worst-case peak, as shown in the following proposition concerning the detection of sinusoids in noise.

Proposition 2.2 (noise tolerance): Let x be a complex sinusoid mixed with noise r, i.e. $x_n=ae^{j(2\pi fn+\varphi)}+r_n$ $(0\leq f<0.5)$, $W(f)$ be the window spectrum, K be a positive integer, then the windowed spectrum X_k has a local peak within K bins from Nf, provided that $|R_k|<0.5aW(0)\Delta$ for $Nf\text{-}1\text{-}K<k<Nf\text{+}1\text{+}K$, where Δ is defined as

$$\Delta = \max_{L \in Z^{+}, L \le K} \Delta_L \,, \quad \Delta_L = \inf_{|f| \le 0.5/N} \frac{|W(f)| - |W(f + L/N)|}{W(0)} \,. \tag{2.2}$$

The proof is given in Appendix B.1.2. Δ_L is interpreted as the minimal drop from the main sampled peak of W to a bin L bins away. If we denote the DFT of the sinusoid (without noise) as S, then the proposition simply says that a local peak is guaranteed for X within K bins from the main peak of S, if the noise is not strong enough to make up the drop from the main peak to somewhere no more than K bins from it. However, the local peak of X does not have to be the main peak of S. The value of Δ depends mainly on the window type, the integer K, and very slightly on the window size N if N is large. For large N numerical results of Δ are given in Figure 2.1.

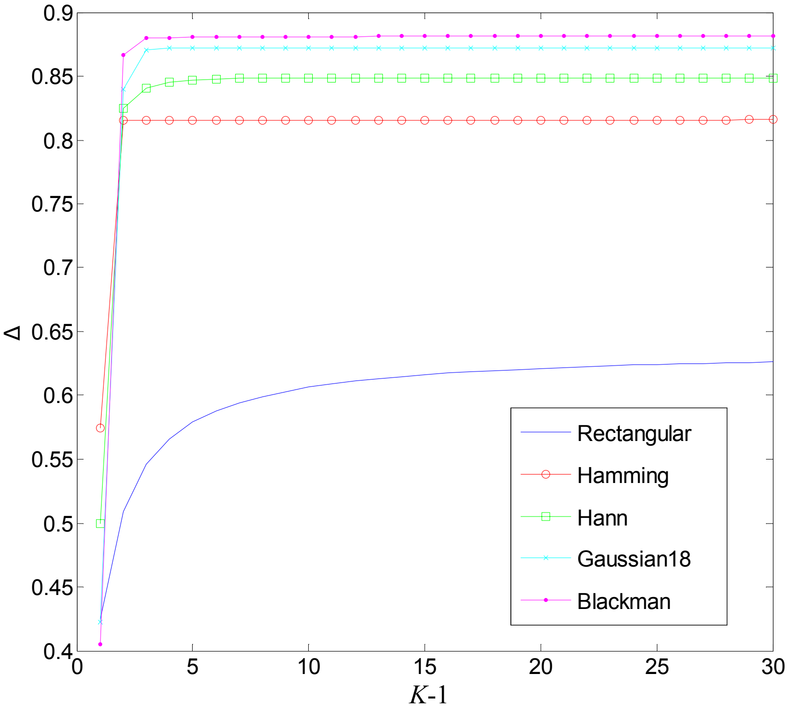


Figure 2. 1 Noise tolerance factor (Δ)

*Gaussian18 is the Gaussian window given as w_n=exp(-18(n-0.5N)$^2N^{-2}$).

All these curves converge to $|W(0.5/N)|/W(0)$ as $K\to\infty$. The rectangular window shows a distinctively lower noise tolerance than the others because its amplitude is lower at the frequency of 0.5 bin. It is also apparent that Δ does not have useful increase after some critical point (2 or 3 bins), around which the integer K is chosen in practice.

Proposition 2.2 takes the deterministic approach with noise. For white noise we get the following result.

Corollary 2.3 (flat-spectrum noise): Let the windowed noise $r{\cdot}w$ have a root-mean-square (r.m.s.) $\sigma_{rw}<0.5aN^{1/2}W(0)\Delta$, its windowed DFT R_k be constant-amplitude, K be a positive integer, Δ be as defined in (2.2), then $x_n=ae^{j(2\pi fn+\varphi)}+r_n$ has a local windowed spectral peak within K bins from Nf.

This is shown by

$$|R_m|^2=N^{-1}\sum_k|R_k|^2=N^{-1}N\sum_n(r_nw_n)^2=N\sigma_{rw}^2\rightarrow|R_m|=N^{1/2}\sigma_{rw}<0.5aW(0)\Delta\,. \quad (2.3)$$

In the above the conditions on r are also deterministic. However, they approximately characterize a white noise with $\sigma_r<0.5a\Delta\|w\|_1/\|w\|_2$, where $\|w\|_1$ and $\|w\|_2$ are the L^1 and L^2 norms, respectively, of w. $\|w\|_1=W(0)$ is proportional to N. $\|w\|_2$ is proportional to $N^{1/2}$. (2.3) shows that longer windows tolerate more wide-band noise.

2.1.2 Two sinusoids

In Proposition 2.2 the noise level is forced under a "flat cap" in the frequency range around the sinusoid, which is chosen to represent wide-band noise. In sinusoid modeling, however, the interference between sinusoids is common. For peak picking involving concurrent sinusoids, we have the following result.

Proposition 2.4 (sinusoidal noise): Let the noise r be a constant sinusoid, i.e. $r_n=be^{j(2\pi gn+\theta)}$, K be a positive integer, then $x_n=ae^{j(2\pi fn+\varphi)}+r_n$ has a local windowed spectral peak within K bins from Nf, provided that $b<a\Delta_s$, where Δ_s is defined as

$$\Delta_s=\max_{L\in Z^+,L\le K}\Delta_s(L),\ \Delta_s(L)=\inf_{|f|\le 0.5/N}\left(\frac{|W(f)|-|W(f+L/N)|}{|W(f\pm h)|+|W(f\pm h+L/N)|}\right),\ h=g\text{-}f. \quad (2.\ 4)$$

The proof is given in Appendix B.1.3. Δ_s depends mainly on the window type, the integer K, the frequency gap h, and very slightly on the window size N if N is large. Figure 3.2 gives numerical results of Δ_s for K=4 and very large N . The curves show local oscillations synchronized to the bins, which is another example of the frequency sampling effect.

The assumptions of Proposition 2.4 is sufficient not only for a peak to exist within K bins from Nf, but also for this peak to be credited to the contribution of the sinusoid at Nf, no matter whether the r has a positive contribution or not. Generally speaking the DFT detector has much higher tolerance of sinusoidal noise than of wide-band noise if the frequency gap h is above several bins, as little noise power spreads to the bins where the sinusoid of interest is.

In practice we always work with real sinusoids rather than complex ones. A real sinusoid is equivalent to two complex ones with conjugate frequencies. Figure 2.2 shows that the existence of a conjugate hardly affects the detectability of a sinusoid, unless the frequency is very low, e.g. below 1 bin.

2.1.3 Zero-padded DFT

In [Serra97] it has been proposed to use zero-padded DFT for sinusoid detection. Zero-padding a sequence of N points to M points ($M>N$) results in a $1/M$-sampling of the DTFT, instead of $1/N$-sampling. We can derive a result similar to Proposition 2.1 by replacing N with M. Another result without involving M in the assumptions is given as follows.

Proposition 2.5 (padded DFT): Let the window spectrum $|W(f)|$ be monotonically decreasing on $[0, 0.5/N]$, and satisfy $|W(0.5/N)|>|W(g)|$, $\forall\, |g|>0.5/N$, x be a constant complex sinusoid, i.e. $x_n=ae^{j(2\pi fn+\varphi)}$ $(0\leq f<1/2)$, X_k be its M-point padded windowed DFT, then X_k has a global peak at bin k, where k/M is closest to f.

Using padded DFT also enhances the noise tolerance by raising the worst-case peak from $W(0.5/N)$ to $W(0.5/M)$. This is easily shown by Proposition 2.2, since zero-padding does not change the DTFT of the window function.

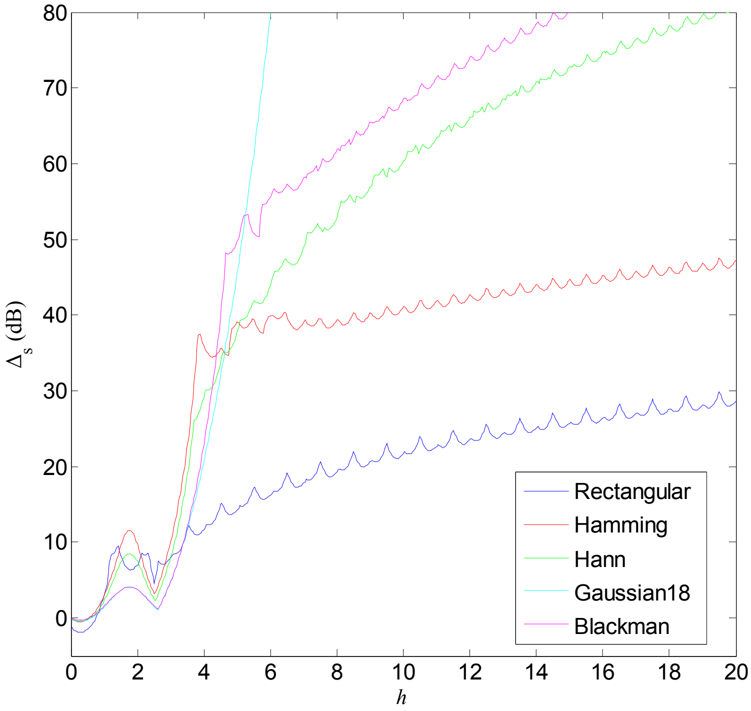


Figure 2. 2 Disturbance tolerance factor Δ_s for K=4

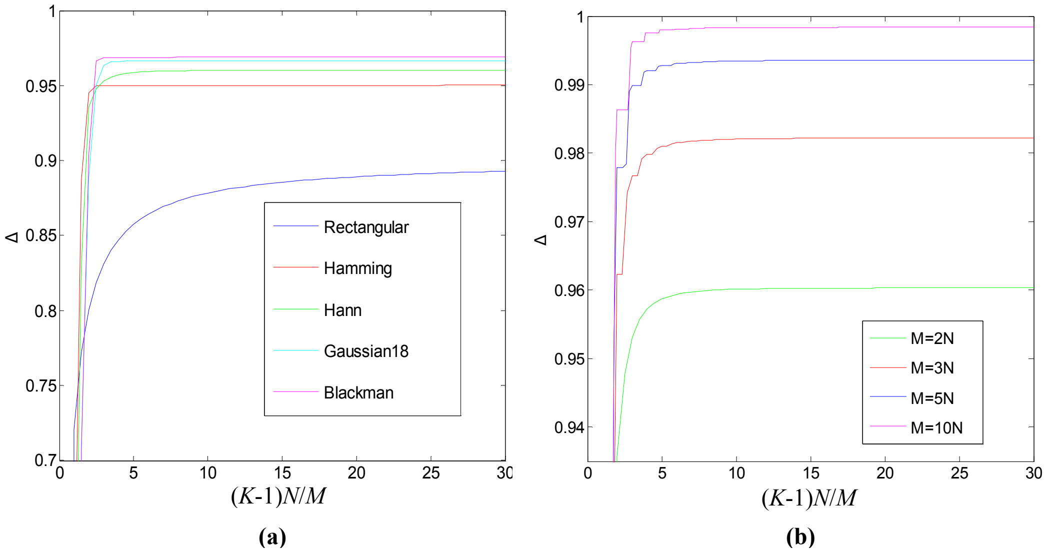


Figure 2. 3 Noise tolerance factor (Δ) using zero-padded DFT

(a) for fixed padding rate (2); (b) for fixed window type (Hann)

Figure 2.3 (a) shows the noise tolerance factor at $M=2N$, N and M being the unpadded and padded window sizes respectively, for various windows. Figure 2.3 (b) compares the noise tolerance for different padding rates computed using the Hann window. In the horizontal axes we have multiplied the number of bins K-1 by N/M to convert it back to the same frequency scale as in Figure 2.1.

2.1.4 Section summary

In §2.1 we have discussed several sufficient conditions for stationary sinusoids to appear in the DFT as spectral peaks, so that they can be detected by direct peak picking. Peak picking is used in standard sinusoid modeling for its simplicity and acceptable performance. For time-varying sinusoids, although it is hard to formulate good bounds regarding noise level or peak position in a general form, peak picking works fine most of the time, as long as the sinusoid is not overwhelmed by other events. However, there is always a chance that a sinusoid, constant or time-varying, does *not* appear as a spectral peak. Special care should be taken of missing peaks in sinusoid or harmonic sinusoid modeling (see §3.2).

2.2 Measuring frequency

Sinusoid modeling systems extract sinusoidal parameters, i.e. frequencies, amplitudes and phase angles, at detected spectral peaks. Although the three parameters are often evaluated together, the estimation of frequency plays a key role, while the estimation of amplitude and phase angle usually depends on the frequency estimate. In this section we focus on frequency estimation only. The amplitude and phase angle are left to §2.3.

Since it is impossible to measure the frequency at a single point with any accuracy, the instantaneous frequency is always measured from a data segment (*frame*) centred at a measurement point, while the segment itself may involve parameter changes. The frequency estimate therefore carries the characteristics of the whole segment, and is almost always inaccurate.

[KM02] summarizes DFT-based methods for estimating stationary sinusoids. These include the standard FFT method, parabolic interpolation [Serra89], reassignment method [AF95], derivative method [DM00], phase vocoder [BP93], etc. All these methods produce one frequency estimate calculated from an interval of N (or a few more) points. In the following we review these methods and derive error estimates in analytical form.

2.2.1 Standard FFT method

This method was suggested in [MQ86]. Given a signal x, the FFT frequency estimator calculates its N-point DFT, then selects local maxima from the amplitude spectrum. Let one of them be selected at bin k, then the frequency is estimated as

$$\hat{f} = k / N \tag{2. 5a}$$

2.2.1.1 Constant sinusoids

For pure complex sinusoids, (2.5a) quantizes the true frequency f to its closest bin k/N. The frequency estimation error is

$$\delta = k / N - f \tag{2. 5b}$$

δ is bounded by 0.5/N. By zero-padding the signal to M points before calculating the DFT, the error bound is reduced to 0.5/M. As $M\to\infty$, $|\hat{f} - f| \leq 0.5/M \to 0$. However, for real sinusoids the error cannot converge to zero due to the existence of a conjugate sinusoid. In this case we derive the following from (1.9) and (1.23):

$$X_k = \frac{a}{2} e^{j\varphi} W(k / N - f)\left(1 + e^{-j2\varphi}\varepsilon\right) \tag{2. 6a}$$

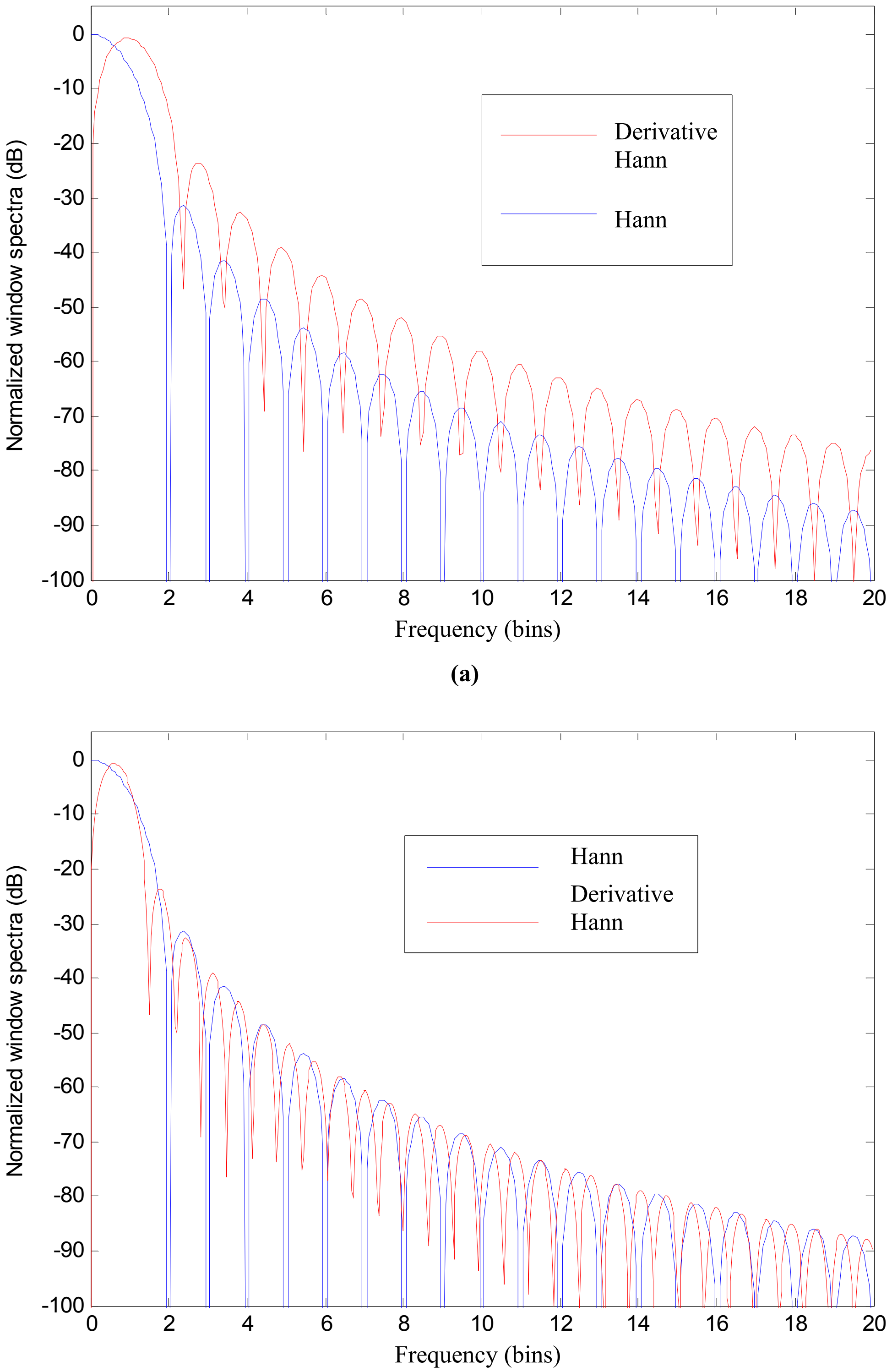
where $\varepsilon = \frac{W(\delta + 2f)}{W(\delta)}$. Since w is low pass, $|\varepsilon|$ is a small number well below 1 when k/N is close to f and f is not too small. For $M\to\infty$, the frequency estimate error is bounded by

$$\left|\hat{f} - f\right| < \frac{\left|W'(\delta + 2f)\right|}{8\pi \sum_{n=-N/2}^{N/2} n^2 w_n} \cdot \frac{1 + |\varepsilon|}{1 - |\varepsilon|} \tag{2. 6b}$$

provided that $\hat{f}$ is detected within 0.5/N from f.

The proof of (2.6b) is given in Appendix B.2.1. $|W'(\delta+2f)|$ is the amplitude spectrum of $2\pi nw_n$ at frequency $\hat{f}+f$, which is within $0.5/N$ from $2f$. For the Hann window, we compare $\Delta(f)=\frac{N|W'(f)|}{8\pi\sum_n n^2 w_n}$ to the normalized window spectrum $\frac{|W(f)|}{\sum_n w_n}$ in Figure 2.4a. $\Delta(f)$ is generally low-pass, but has a notch at zero frequency. As f grows $\Delta(f)$ approaches zero. There is a 39dB decay at 5 bins, and 60dB decay at 11 bins. The decay rate is slower than that of the Hann window spectrum, but the decay law is similar. To show this we compress $\Delta(f)$ along the f axis by 25:16, and compare it to the Hann window spectrum in Figure 2.4b. The two are shown to have highly consistent profiles. In Figure 2.5 we compare $\Delta(f)$ for four window types, namely the Hamming, Hann, Blackman and Gaussian-18 windows. The Hann and Blackman windows show faster decay rate than the other two.

For numerical example, if the signal frequency f is above 5.5bins, then according to (2.6b) the frequency estimation error using padded Hann-windowed DFT is roughly bounded by $0.001/N$ when $M\to\infty$.



(a)

(b)

Figure 2. 4 Comparing Hann window spectrum and its derivative

(a) original spectra; (b) after scaling the derivative window spectrum

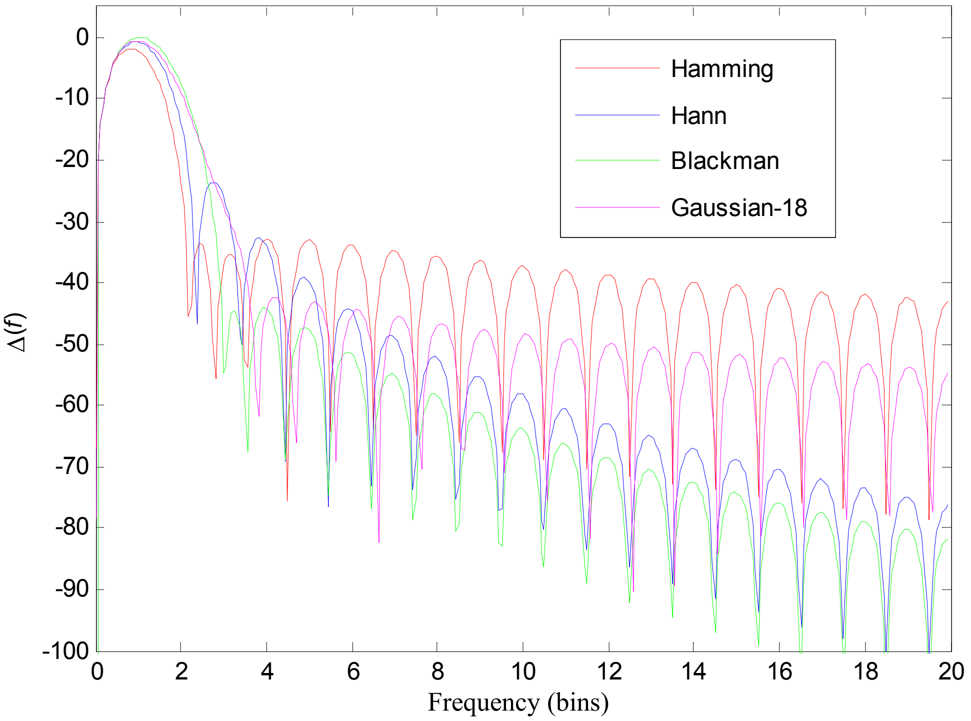


Figure 2. 5 Comparing window spectrum derivatives

2.2.1.2 Time-varying sinusoids

Now suppose we have a time-varying sinusoid $x_n = a_n e^{j(\varphi_0 + 2\pi\int_0^n f(t)dt)}$. Let the spectral peak be located at $\hat{f}$ when $M\to\infty$, we derive the following

$$\hat{f} = \frac{\sum_{n=0}^{N-1}\sum_{m=0}^{N-1} w_{mn} a_m a_n \operatorname{sinc}\frac{\Delta\varphi_{mn}}{\pi}\cdot\int_m^n f(t)dt}{\sum_{n=0}^{N-1}\sum_{m=0}^{N-1} w_{mn} a_m a_n \operatorname{sinc}\frac{\Delta\varphi_{mn}}{\pi}\cdot(n-m)} \qquad (2.\ 7a)$$

or

$$\hat{f} = \frac{\sum_{l=0}^{N-2}\eta_l\int_0^1 f(l+t)dt}{\sum_{l=0}^{N-2}\eta_l},\ \eta_l = \sum_{n=l+1}^{N-1}\sum_{m=0}^{l} w_{mn} a_m a_n \operatorname{sinc}\frac{\Delta\varphi_{mn}}{\pi} \qquad (2.\ 7b)$$

where $\Delta\varphi_{mn}=\varphi_n-\varphi_m-2\pi\hat{f}(n-m)$, $w_{mn}=(n-m)w_m w_n$.

The details of deriving (2.7a) and (2.7b) are given in Appendix B.2.2. (2.7b) expresses the DTFT peak frequency as a weighted average of the instantaneous frequency, with η_l being the weight on the interval [l, l+1]. However, it is not a closed-form solution, because the right hand side depends on $\hat{f}$ itself.

In most DFT-based estimators the frequency estimate is assigned to the frame centre (with the exception of time-reassignment [AF95]). Therefore we compare the frequency estimate to the instantaneous frequency at the central point N/2 (sometimes we take (N-1)/2) to get an estimation error. This is given as

$$f_0 - f_{N/2} = \frac{\sum_{l=0}^{N-2} \eta_l \left(\varphi_{l+1} - \varphi_l - f(N/2) \right)}{\sum_{l=0}^{N-2} \eta_l} \qquad (2.\ 7c)$$

2.2.2. Parabolic interpolation

The parabolic method was used in [Serra89]. Although the standard FFT method can achieve arbitrarily high frequency accuracy by zero padding, the additional computation cost prevents us doing so with a high padding-rate. Since zero-padding does nothing beyond a denser sampling of the DTFT, and since the DFT calculated with any zero-padding rate contains the complete information, it has been suggested that we may interpolate the spectrum calculated at a relatively low zero-padding rate to achieve a high frequency resolution. The parabolic interpolation is the simplest case of such interpolations.

Like the standard FFT method, the parabolic interpolation first locates a local spectral peak at bin k, so that $|X_k|\geq|X_{k-1}|$, $|X_k|\geq|X_{k+1}|$. The amplitude spectrum (direct or post-processed) between bins k-1 and k+1 is then approximated with a quadratic function, then the peak position of this quadratic function is used as the frequency estimate. Let these three amplitudes be A_{-1}, A_0, and A_1, respectively. The quadratic function is given as

$$A(k+d) = (0.5A_{-1} - A_0 + 0.5A_1)d^2 - 0.5(A_{-1} - A_1)d + A_0 \qquad (2.\ 8a)$$

The local maximum is found at

$$d = \frac{1}{2}\frac{A_{-1} - A_1}{A_{-1} - 2A_0 + A_1} \tag{2. 8b}$$

Finally the frequency is estimated as

$$\hat{f} = \frac{k+d}{M} \tag{2. 8c}$$

where M is the zero-padded DFT size. For a pure complex sinusoid the amplitude spectrum is given by $|X_k|=|W(k/M-f)|$. Let k be the integer closest to Mf, i.e. $|k/M-f|\leq 0.5/M$; δ be $k/M-f$. Without loss of generality, let $k/M>f$. Then

$$A_{-1}=|W(\delta-1/M)|,\ \mathrm{A}_0=|W(\delta)|,\ \mathrm{A}_1=|W(\delta+1/M)|, \tag{2. 9a}$$

so

$$d = \frac{1}{2}\frac{|W(\delta-1/M)| - |W(\delta+1/M)|}{|W(\delta-1/M)| - 2|W(\delta)| + W|(\delta+1/M)|}. \tag{2. 9b}$$

The frequency estimation error is

$$\hat{f} - f = \frac{k+d}{M} - \left(\frac{k}{M} - \delta\right) = \frac{d}{M} + \delta \tag{2. 9c}$$

The parabolic method assumes d approximates $-\delta M$. Generally speaking the performance of this method depends on how accurately W can be approximated by a parabolic function near zero frequency. When $|W|$ is quadratic near zero, i.e. $|W(f)|=a+bf^2$, we can verify that $d=-\delta M$. In a more general case, let $|W(f)|$ be approximated around zero as

$$|W(f)| = \sum_{n\geq 0} A_n (Nf)^{2n}, \tag{2. 10a}$$

where $N\leq M$ is the window length without padded zeros, then we may calculate the frequency error in bins

$$d + \delta M = \frac{1}{2}\frac{\sum_n A_n (N/M)^{2n} I_n(\delta M)}{\sum_n A_n (N/M)^{2n} J_n(\delta M)} \tag{2. 10b}$$

where

$$I_n(x) = (1+2x)(x-1)^{2n} + (-1+2x)(x+1)^{2n} - 4x(x)^{2n} \tag{2. 10c}$$

$$J_n(x) = (x-1)^{2n} - 2(x)^{2n} + (x+1)^{2n} \tag{2. 10d}$$

I_n and J_n have the following properties:

1) $I_0(x)=I_1(x)=J_0(x)=0$;

2) $I_n(x)=-I_n(-x)$, $J_n(x)=J_n(-x)$;

3) $I_n(0)=I_n(0.5)=0$;

4) $I_n(x)<0$ when $n\geq 2$ and $0<x<0.5$, $J_n(x)>0$ when $n\geq 1$;

5) $I_n(x)/J_n(x)>-1$ when $n\geq 2$ and $0<x<0.5$ (so that $|I_n(x)|<|J_n(x)|$);

6) $J_n(x)\leq 1+(x+1)^{2n}$, and grows with n like $(1+x)^{2n}$ when $0<x<0.5$.

First part of 4) is shown by the following induction:

$$\begin{cases} I_1(x) = 0; \\ \text{If } I_{n-1}(x) \leq 0, \text{ then } I_n(x) = (x-1)^2 I_{n-1}(x) + 4x(-1+2x)\left((x+1)^{2(n-1)} - x^{2(n-1)}\right) < 0 \end{cases}$$

To show 5) we write

$$\frac{I_n(x)}{J_n(x)} + 1 = \frac{2}{J_n(x)}\left((1+x)(1-x)^{2n} + x(1+x)^{2n} - (1+2x)x^{2n}\right) = \frac{2}{J_n(x)}K_n(x).$$

We show the above is positive by the following induction:

$$\begin{cases} K_1(x) = 1; \\ \text{If } K_{n-1}(x) \geq 0, \text{ then } K_n(x) = (1-x)^2 K_{n-1}(x) + 4x^2(1+x)^{2(n-1)} + (1-4x^2)x^{2(n-1)} > 0 \end{cases}$$

With “normal” low-pass window functions A_n decreases rapidly with n. As J_n increases like 1.5^{2n} at the most, and $|I_n|$ is below $|J_n|$, when the zero-padding rate M/N is moderately high, e.g. $M=10N$, the factor $(N/M)^{2n}$ assures that the error is dominated by the first non-zero terms in both the numerator and denominator of (2.10b). In particular, if $A_1\neq 0$, $A_2\neq 0$, we can write

$$d + \delta M \cong \frac{1}{2}\left(\frac{N}{M}\right)^2 \frac{A_2 I_2(\delta M)}{A_1 J_1(\delta M)} = \frac{N^2}{2M^2}\frac{A_2}{A_1}\frac{-4\delta M + 16(\delta M)^3}{2}. \tag{2. 10e}$$

As $\left|\frac{-4\delta M+16(\delta M)^3}{4}\right|$ is bounded by $3^{-1.5}$, the frequency error is roughly bounded by $\frac{N^2}{3^{1.5}M^3}\left|\frac{A_2}{A_1}\right|$.

For numerical example, we look at the Hann window, approximating its window spectrum with that of a continuous Hann window. We calculate

$$\begin{aligned}|W(f)| &\cong 0.5N\left|\left(\operatorname{sinc}(Nf)+0.5\operatorname{sinc}(Nf-1)+0.5\operatorname{sinc}(Nf+1)\right)\right| \\ &= 0.5N\cdot\left(\operatorname{sinc}(Nf)+0.5\operatorname{sinc}(Nf-1)+0.5\operatorname{sinc}(Nf+1)\right) \quad \left(|Nf|\le 2\right) \qquad (2.\ 10f) \\ &= 0.5N\cdot\frac{\sin\pi Nf}{\pi Nf\left(1-(Nf)^2\right)} = 0.5N\cdot\left(\sum_{n=0}^{\infty}\frac{(-1)^n}{(2n+1)!}(\pi Nf)^{2n}\right)\left(\sum_{m=0}^{\infty}(Nf)^{2m}\right) \quad \left(|Nf|\le 1\right)\end{aligned}$$

Comparing this to $W(f)=\sum_{n\ge 0}A_n(Nf)^{2n}$ we get

$$A_n = 0.5N\cdot\sum_{m=0}^{n}\frac{(-1)^m}{(2m+1)!}\pi^{2m} \qquad (2.\ 10g)$$

n	$A_n/0.5N$
0	1
1	-0.644934
2	0.166808
3	-0.239435×10^{-1}
4	0.220438×10^{-2}
5	-0.141699×10^{-3}
6	0.672989×10^{-5}
7	-0.245985×10^{-6}
8	0.713635×10^{-8}
9	-0.168360×10^{-9}
10	0.329383×10^{-11}

Table 2. 1 Polynomial coefficients of Hann window spectrum

The values for n=0~10 are listed in Table 3.1. We have $3^{-1.5}|A_2/A_1|\approx 0.05$.

The error bound above is derived for pure complex sinusoids. In the case of real sinusoids, the existence of the conjugate sinusoid introduces an additional term to the error bound, given by (2.6b) as $M\to\infty$.

2.2.3. Reassignment method

The original reassignment method [AF95] is based on the idea of undoing the smoothing of Wigner-Ville distributions, which generates time-frequency representations such as the short-time Fourier transform. [KM02] takes a stationary assumption and ignores the time reassignment. The frequency reassignment is calculated from two DFTs using window derivative. Let x be a sinusoid, w be a window function with derivative w', the frequency reassignment method calculates the DFT X^{w}, with window w, and $X^{w'}$ with window w', then estimates the frequency as

$$\hat{f} = \frac{k}{N} - \frac{1}{2\pi}\mathrm{Im}\frac{X_k^{w'}}{X_k^{w}}. \qquad (2.\ 11)$$

where k is a local spectral peak. In case of multiple sinusoids, each frequency is estimated using a distinct k.

In this method w' is taken as the derivative of the continuous window function w, although in the calculation of DFT both w and w' are sampled. Let $W_c(f)$ be the spectrum of the continuous window w (not a DTFT), δ=k/N-f, then the frequency estimation error for a pure complex sinusoid x_n=$a\cos^{j2\pi fn}$ is given as

$$\hat{f} - f = \frac{\sum_{n=-\infty}^{\infty} n\cdot W_c(\delta-n)}{\sum_{n=-\infty}^{\infty} W_c(\delta-n)} \qquad (2.\ 12)$$

The proof is given in Appendix B.2.3. (2.12) shows that using a derivative window we can estimate the frequency without error, provided that $W_c(f)$ is band-limited to (-1/2, 1/2). However, no window function has its frequency and time support both bounded. In the case of DFT, since $w(\mathrm{t})$ has bounded time support, $\mathrm{W_c}(f)$ always has marginal energy outside (-1/2, 1/2).

As an example we look at the Hann window in the continuous form

$$w(t) = 0.5\int_{-N/2}^{N/2}(1+\cos 2\pi t/N)dt \tag{2. 13a}$$

with the Fourier transform

$$W_c(f) = \begin{cases} 0.5N, \quad f = 0 \\ 0.25N, \quad f = \pm 1/N \\ -\dfrac{\sin \pi N f}{2\pi N^2 f(f-1/N)(f+1/N)}, \quad \textit{otherwise} \end{cases} \tag{2. 13b}$$

When N is a large even integer, $n\neq 0$ and $f<<1$, we have

$$\begin{aligned} W_c(f+n) &= -\frac{\sin \pi N(f+n)}{2\pi N^2(f+n)(f+n-1/N)(f+n+1/N)} \\ &\cong W_c(f)\frac{f(f-1/N)(f+1/N)}{n^3} \end{aligned} \tag{2. 13c}$$

The frequency estimation error is roughly

$$\begin{aligned} \hat{f}-f &\cong \frac{\sum_{n\neq 0} n\cdot W_c(\delta)\delta(\delta-1/N)(\delta+1/N)(-n)^{-3}}{W_c(\delta)+\sum_{n\neq 0} W_c(\delta)\delta(\delta-1/N)(\delta+1/N)\cdot(-n)^{-3}} \\ &= \frac{-\delta(\delta-1/N)(\delta+1/N)\sum_{n\neq 0} n^{-2}}{1-\delta(\delta-1/N)(\delta+1/N)\sum_{n\neq 0} n^{-3}} \\ &\cong -\delta(\delta-1/N)(\delta+1/N)\left(2\sum_{n=1}^{\infty} n^{-2}\right) = \frac{\pi^2\delta(1-\delta N)(\delta N+1)}{3N^2} \end{aligned} \tag{2. 13d}$$

As $|\delta N(1\text{-}\delta N)(1+\delta N)|$ is bounded by $2\cdot 3^{-1.5}$ when $|\delta N|<1$, we see that the frequency error has an upper bound close to $2\cdot 3^{-2.5}\pi^2/N^2$ bins, if k is selected within 1 bin from Nf.

For real sinusoids, the error (2.12) is replaced by the following:

$$\hat{f} - f = \mathrm{Re}\frac{\sum_{n=-\infty}^{\infty} n \cdot W_c(\delta - n) + e^{-j2\varphi} \sum_{n=-\infty}^{\infty} (-2f + n) \cdot W_c(\delta + 2f - n)}{\sum_{n=-\infty}^{\infty} W_c(\delta - n) + e^{-j2\varphi} \sum_{n=-\infty}^{\infty} W_c(\delta + 2f - n)} \tag{2. 14a}$$

where φ is the phase angle at the centre of the window. When the frequency is low, the above can be approximately written as

$$\hat{f} - f \cong -\mathrm{Re}\frac{e^{-j2\varphi} 2f \cdot W(\delta + 2f)}{W(\delta) + e^{-j2\varphi} W(\delta + 2f)}, \tag{2. 14b}$$

so that the error bound is roughly given by

$$\left|\hat{f} - f\right| < C \cong 2f \cdot \frac{|\varepsilon|}{1 - |\varepsilon|} \tag{2. 14c}$$

where $\varepsilon = \frac{W(\delta + 2f)}{W(\delta)}$.

(2.14c) indicates that one one should select *k* close to the spectral peak, so that *W*(δ) is kept small, to get good estimation.

2.2.4. Derivative method

The name “derivative” originates from the continuous version of the method. In the discrete case the derivative is replaced by the difference. Let *x* be a constant sinusoid, i.e. $x = ae^{j2\pi fn + \varphi}$, Δx be its backward difference, X^0 and X^1 be their DFTs respectively, *w* be a low-pass window function, then the frequency is estimated as

$$f = \frac{1}{\pi}\arcsin\left(\frac{1}{2}\left|\frac{X_k^1}{X_k^0}\right|\right) \tag{2. 15}$$

where *k* is a spectral peak of X^0. In case of multiple sinusoids, each frequency is estimated using a distinct *k*.

The derivative method does not have an error for pure complex sinusoids. For real sinusoids and not too low *f*, the error bound is given by

$$\left|\hat{f}-f\right| \leq \frac{2\left|\tan \pi f\right|}{\pi}|\varepsilon| \tag{2. 16}$$

where $\varepsilon=\frac{W(\delta+2f)}{W(\delta)}$, δ=*k*/*N*-*f*. The proof is given in Appendix B.2.4.

(2.16) indicates that one one should select *k* close to the spectral peak, so that *W*(δ) is kept small, to get good estimation.

2.2.5. Phase vocoder method

The phase vocoder method [BP93] estimates the frequency using phase changes of the Fourier transform. Let $x=ae^{j2\pi fn+\varphi}$, *w* be a low-pass window function, *X* be the windowed DFT of *x* calculated from *N* points 0, 1, …, *N*-1, X^1 be the windowed DFT of *x* calculated from *N* points 1, 2, …, *N*. The phase vocoder estimates the frequency as

$$\hat{f}=\frac{1}{2\pi}\arg\frac{X_k^1}{X_k} \tag{2. 17}$$

where the arg function takes the angle of a complex number between –π and π. The method is also known as phase difference method as $\arg\frac{X_k^1}{X_k}=\arg X_k^1-\arg X_k$. Like the derivative method, the phase vocoder method does not have an error for pure complex sinusoids. For real sinusoids and not too low *f*, the frequency estimation error is bounded by

$$\left|\hat{f}-f\right| \leq 2\arcsin|\varepsilon| \tag{2. 18}$$

where $\varepsilon=\frac{W(\delta+2f)}{W(\delta)}$, δ=*k*/*N*-*f*. The proof is given in Appendix B.2.5. (2.18) indicates that one one should select *k* close to the spectral peak, so that *W*(δ) is kept small, to get good estimation.

2.2.6. Section summary

In §2.2 we have discussed several DFT-based methods for frequency estimation. Although the standard DFT only provides a frequency resolution of 1/*N*, almost all these DFT-based methods can achieve much higher accuracy for clean and constant sinusoids. Of the five methods, the standard FFT method requires the calculation of heavily padded DFT to achieve good result; the reassignment, derivative and phase vocoder methods require computing two FFTs; the parabolic interpolation method requires only one FFT. We compare the error bounds of four methods for constant real sinusoid in Table 2.2. The parabolic method has the error bound of the heavily-zero-padded FFT method, plus an extra term due to non-parabolicity, which vanishes if a window function with parabolic amplitude spectrum near zero frequency is used.

Method	Error bound
Standard FFT (with heavy zero padding)	$\left\|\hat{f}-f\right\| < \frac{\left\|W'(\delta+2f)\right\|}{8\pi \sum_{n=-N/2}^{N/2} n^2 w_n} \cdot \frac{1+\|\varepsilon\|}{1-\|\varepsilon\|}$
Reassignment	$\left\|\hat{f}-f\right\| < C \cong 2f \cdot \frac{\|\varepsilon\|}{1-\|\varepsilon\|}$
Derivative	$\left\|\hat{f}-f\right\| \le \frac{2}{\pi}\left\|\tan \pi f\right\|\left\|\varepsilon\right\|$
Phase vocoder	$\left\|\hat{f}-f\right\| \le 2\arcsin\|\varepsilon\|$

Table 2. 2 Comparing error bounds for pure sinusoids

2.3. Measuring amplitude and phase angle

Given (1.23), the amplitude and phase angle of a constant sinusoid can be straightforwardly estimated as

$$\hat{a} = \left|\frac{X_k}{W(k/N-\hat{f})}\right|, \quad \hat{\varphi} = \arg\frac{X_k}{W(k/N-\hat{f})} \tag{2. 19}$$

This method is used in [AKZ99] and [KM02] as an improvement on former amplitude and phase estimation methods proposed in conjunction with individual frequency estimation methods. For a pure complex sinusoid, if the frequency is determined without error, then (1.23) assures that the amplitude and phase angles can also be. In the case of real sinusoids, we have

$$X_k = \frac{a}{2}\left(e^{j\varphi}W(k/N-f)+e^{-j\varphi}W(k/N+f)\right), \tag{2. 20a}$$

the relative amplitude error

$$\left|\frac{\hat{a}-a/2}{a/2}\right| = \left|\left|\frac{e^{j\varphi}W(\delta)+e^{-j\varphi}W(\delta+2f)}{W(\delta)}\right|-1\right| = \left|\left|1+e^{-j2\varphi}\varepsilon\right|-1\right| \le |\varepsilon| \tag{2. 20b}$$

and the phase angle error

$$|\hat{\varphi}-\varphi| = \left|\arg e^{j\varphi}\left(1+e^{-j2\varphi}\varepsilon\right)-\varphi\right| = \left|\arg\left(1+e^{-j2\varphi}\varepsilon\right)\right| \le \arcsin|\varepsilon| \tag{2. 20c}$$

where $\varepsilon = \dfrac{W(\delta+2f)}{W(\delta)}$, $\delta = k/N - f$.

However, the frequency estimate almost always has an error. Let the frequency estimate be $\hat{f}$ and $\delta_f = \hat{f} - f$. Then for a pure complex sinusoid, the amplitude estimation error is

$$\begin{aligned} \left|\frac{\hat{a}-a}{a}\right| &= \left|\left|\frac{e^{j\varphi}W(\delta)}{W(\delta-\delta_f)}\right|-1\right| = \left|\left|1+\frac{W(\delta)-W(\delta-\delta_f)}{W(\delta-\delta_f)}\right|-1\right| \\ &\le \left|\frac{W(\delta)-W(\delta-\delta_f)}{W(\delta-\delta_f)}\right| \le \frac{\sup\limits_{0\le\theta\le1}\left|W'(\delta-\theta\delta_f)\right|}{\left|W(\delta-\delta_f)\right|}\left|\delta_f\right| \end{aligned} \tag{2. 21a}$$

According to (2.21a), the frequency estimation error introduces an amplitude error that depends on the behaviour of *W*(*f*) near *f*. For a real sinusoid the conjugate sinusoid introduces a similar term as in (2.20b):

$$\left|\frac{\hat{a}-a/2}{a/2}\right| = \left|\left|\frac{e^{j\varphi}W(\delta)+e^{-j\varphi}W(\delta+2f)}{W(\delta-\delta_f)}\right|-1\right|$$

$$= \left|\left|1+\frac{W(\delta)-W(\delta-\delta_f)}{W(\delta-\delta_f)}+e^{-j2\varphi}\varepsilon\left(1+\frac{W(\delta)-W(\delta-\delta_f)}{W(\delta-\delta_f)}\right)\right|-1\right| \qquad (2.\ 21b)$$

$$\le |\varepsilon| + (1+|\varepsilon|)\left|\frac{W(\delta)-W(\delta-\delta_f)}{W(\delta-\delta_f)}\right| \le |\varepsilon| + \frac{\sup\limits_{0\le\theta\le1}|W'(\delta-\theta\delta_f)|}{|W(\delta-\delta_f)|}(1+|\varepsilon|)|\delta_f|$$

However, the frequency error does not affect the phase estimate, unless the error is so large that $W_k(f)$ and $W_k(f+\delta)$ have different signs.

In practice W' is of the same magnitude as W itself when the frequency is expressed in bins, so if δ_f is on the magnitude of bins, the amplitude error bound is on the magnitude of 1, i.e. very large. Therefore to use (2.19) for amplitude estimation requires a highly accurate frequency estimate, which is not easily available for fast varying frequencies. Accurate frequency estimation involving frequency variation model will be discussed in §3.3.

2.4 Sinusoid tracking methods

In sinusoid modeling, the tracking stage connects short-time sinusoid atoms into sinusoid tracks, along which the parameters evolve. All criteria for connecting sinusoids from one point to the next are based on some kind of continuity measure: only those atoms that bear some similarity are connected together.

The first and most important continuity measure is the closeness in frequency. In [MQ86] the authors composed their sinusoid tracking using this measure alone. A track is extended forward if there is a sinusoid atom ahead and its frequency is within a maximal jump from the frequency at the end of the current track. If more than one atom is found, the one with the smallest frequency jump is selected. A track “dies” if no atom exists in the allowed range. Atoms that are not matched to existing tracks mark the “birth” of new tracks. Figure 3.4, taken from [Serra97], illustrates this partial tracking. Compared to [MQ86], [Serra97] allows a dead track to be revived if a successor can be found within a short time.

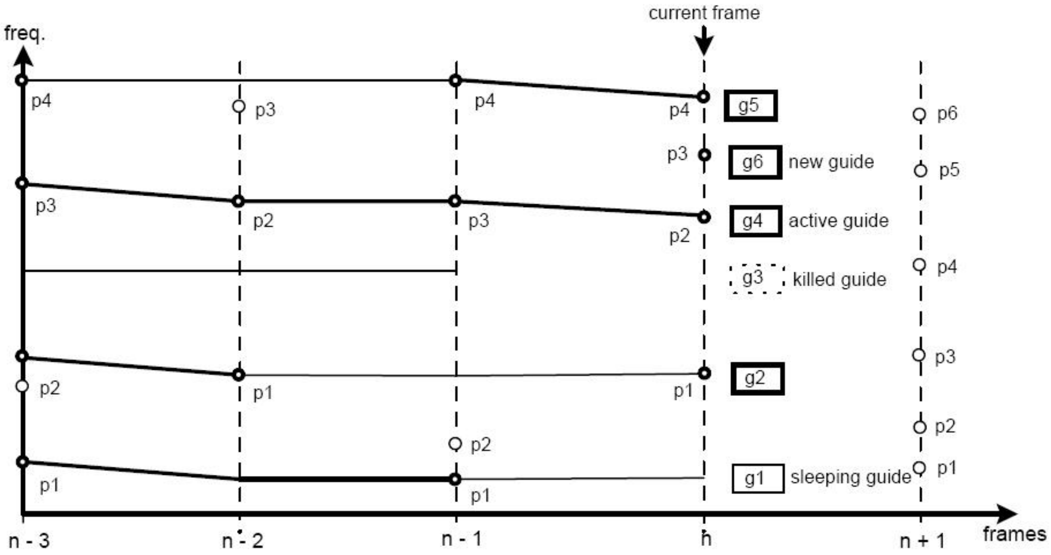


Figure 2. 6 Partial tracking by frequency closeness [Serra97]

Advancements in partial tracking have mainly focused on deriving patterns for frequency and amplitude variation. [Röbel02] proposed to estimate the frequency slope using the reassignment method and suggested using this slope in partial tracking. Instead of directly evaluating the frequency slope, [SW98] implements a linear frequency predictor as a Kalman tracker, with the frequency slope as a state variable. The use of frequency variation trends for guiding partial tracking appears in [LMRR03] as a linear prediction model, exploring higher-order trends than a simple first-order slope. The linear prediction model is able to closely model exponential chirps and sinusoidal modulations. The predictor for a certain frame is calculated from a chosen number of previous frames. All the above methods are examples of pure forward tracking. One drawback to pure forward or backward tracking is the sensibility to local disturbance. On the contrary, forward-backward partial tracking improves the robustness by introducing a global cost function. [SB90] and [SE90] are early examples of using hidden Markov model for tracking sinusoids. In the context of sinusoid modeling, [DGR93] explores the use of hidden Markov model for modeling first-order amplitude and frequency trends. Compared to forward frequency prediction models, the use of a forward-backward tracking framework makes it possible to find optimal partials globally.

All the methods above deal with tracking individual partials. Most existing methods that model harmonic sources [BC94, Ellis96, Métois98, Tolonen99] do so by tracking individual partials then grouping them according to some harmonic relationship. [BC94, Ellis96] describe general-purpose audio analyzers called computational auditory scene analysis (CASA) systems. These systems do not model sinusoids explicitly, but they collect energy in the time-frequency space into partial tracks in a manner similar to sinusoid modeling. [Tolonen99] applies the harmonic criterion after a sinusoid modeling stage to form harmonic streams. There are also methods that consider harmonic constraints in the first place. [Brown92] introduces a pattern for harmonic particles that explicitly requires that the objects being tracked be harmonic. [DG03] models partial harmonicity by forcing partial frequencies close to perfect harmonicity in formulating a Bayesian model. Amplitude variation is modeled as "smooth" in [DG03] by forcing the amplitude to be an additive combination of low-pass envelopes. Frequency variation modeled in the Bayesian framework in [WGR99] using single-frame harmonic models connected in a Markov chain. [DD07] formulates the detecting and tracking of harmonic sinusoids in a Bayesian sequential harmonic framework, which models a wider range of signals than previous Bayesian harmonic methods.

2.5 Synthesizing methods

Given the amplitude a and phase angle φ, a sinusoid can be synthesized by directly calculating $x_n = a_n \cos\varphi_n$. However, in frame-based modeling the parameters are only estimated at the frame centres. The synthesizer therefore has to recover the skipped points from the estimates. The first sinusoid synthesizers that rebuild sinusoids from incomplete parameter estimates were introduced in [MQ86]. The first method proposed by [MQ86] uses an overlap-add framework. Rather than reconstructing the missing parameters, the overlap-add method directly interpolate sinusoid atoms linearly. This interpolation is not consistent with the sinusoid model, but has good perceptual quality when the overlap rate is high. The overlap-add synthesis can be implemented by fast Fourier transform [RD92].

[MQ86] also gives a true sinusoid synthesizer that rebuilds the signal by recovering instantaneous parameters at skipped points. The parameters are interpolated between adjacent pairs of measurement points using the estimates at these two points only. Let them be at time 0 and *N*, with parameter estimates ($\hat{a}_0$, $\hat{f}_0$, $\hat{\varphi}_0$) and ($\hat{a}_N$, $\hat{f}_N$, $\hat{\varphi}_N$) respectively. The parameters between 0 and *N* are interpolated as

$$\tilde{a}_n = \hat{a}_0 + \frac{n}{N}(\hat{a}_N - \hat{a}_0),\ \tilde{\varphi}_n = \hat{\varphi}_0 + 2\pi\left(\hat{f}_0 n + \frac{\hat{f}_N - \hat{f}_0}{2N}n^2 + \alpha(3N-2n)n^2\right) \quad (2.\ 22)$$

where *α* is chosen to be the smallest number that makes $\varphi_N - \hat{\varphi}_N$ a multiple of 2*π*. In the MQ synthesizer the amplitude is piecewise linear, while the phase angle is piecewise cubic. The phase interpolation suffers from phase wrapping around 2*π*. In the original article the authors derived the phase interpolation algorithm by minimizing the frequency fluctuation under four boundary conditions, using an argument counting the number of phase wraps. However, in (2.22) we have formulated it as a linear frequency interpolation with a minimal correction term. It is apparently that (2.22) satisfies the four phase and frequency boundary conditions, i.e. $\tilde{\varphi}_0 = \hat{\varphi}_0$, $\tilde{\varphi}_N - \hat{\varphi}_N = 2k\pi$, $\tilde{\varphi}'(t)|_{t=0} = 2\pi\hat{f}_0, \tilde{\varphi}'(t)|_{t=N} = 2\pi\hat{f}_N$. The frequency fluctuation, as given by [MQ86], is

$$\int_0^N (\tilde{\varphi}''(t))^2 dt = 4\pi^2 \int_0^N \left(\frac{\hat{f}_N - \hat{f}_0}{N} + 6\alpha(N-2t)\right)^2 dt = \frac{4\pi^2\left(\hat{f}_N - \hat{f}_0\right)^2}{N} + 48\pi^2 N^3 \alpha^2 \quad (2.\ 23)$$

This shows that by choosing a minimal coefficient α for the correction term in (2.22), we also minimize the frequency fluctuation.

There have been variations of the MQ synthesizer, with boundary conditions different from the standard MQ synthesis. [Serra97] gives a simplified version that interpolates frequency linearly and discards phase estimates. Since the human auditory system is not sensitive to phase angles, discarding the phase of stationary sinusoids does not undermine the perceptual quality of the synthesized sound. On the contrary, [LMMRP03] proposes to introduce more boundary conditions for enhanced accuracy. The original boundary conditions of the MQ synthesizer assure the 1st- and

2nd-order continuity of phase angle. In [LMMRP03] the conditions are extended to ensure the 3rd-order continuity, provided that estimates of the frequency derivative are available. Indeed, with frequency and frequency derivative estimated at time 0 and *N* as $\hat{f}_0$, $\hat{f}'_0$, $\hat{f}_N$, $\hat{f}'_N$, the frequency can be interpolated as the trinomial

$$\tilde{f}^\circ(t)=\hat{f}_0\frac{(t-N)^2}{N^2}\left(1+2\frac{t}{N}\right)+\hat{f}'_0\frac{t(t-N)^2}{N^2}+\hat{f}_N\frac{t^2}{N^2}\left(1-2\frac{t-N}{N}\right)+\hat{f}'_N\frac{(t-N)t^2}{N^2}$$

(2. 24a)

Now we formulate the phase unwrapping problem considering frequency derivative in terms of interpolation and correction. In a wider sense, let $\tilde{f}^\circ(t)$ be any function that is continuous and has continuous derivatives at 0 and *N*. The corresponding phase angle is

$$\tilde{\varphi}_n^\circ=\hat{\varphi}_0+2\pi\int_0^n\tilde{f}^\circ(t)dt \qquad (2.\ 24b)$$

$\tilde{\varphi}_n^\circ$ satisfies the phase continuity condition at boundary 0, i.e. $\tilde{\varphi}_0^\circ=\hat{\varphi}_0$. The continuity at boundary *N* can be satisfied by introducing a correction term *δ*, so that $\delta_0=\delta'_0=\delta'_N=\delta''_0=\delta''_N=0$, $\tilde{\varphi}_N^\circ+\delta_N=\hat{\varphi}_N+2k\pi$. *δ* does not affect the already satisfied boundary conditions. The simplest way to construct *δ* is using a 5th-order polynomial

$$\delta_n=\delta_N\frac{n^3}{N^3}\left(10-15\frac{n}{N}+6\frac{n^2}{N^2}\right) \qquad (2.\ 24c)$$

This correction term is proportional to δ_N. In the minimal correction criterion, we choose δ_N to be the smallest value that satisfies $\tilde{\varphi}_N^\circ+\delta_N=\hat{\varphi}_N+2k\pi$, then interpolate the phase angle as

$$\tilde{\varphi}_n=\tilde{\varphi}_n{}^\circ+\delta_n \qquad (2.\ 24d)$$

We can also find the instantaneous frequency of the reconstructed sinusoid as

$$\tilde{f}(t)=\tilde{f}^\circ(t)+\frac{\delta_N}{\pi}\frac{15}{N^3}\left(1-\frac{2}{N}t+\frac{1}{N^2}t^2\right)t^2 \qquad (2.\ 24e)$$

2.6 From sinusoid model to harmonic sinusoid model

Sinusoid modeling assumes that an audio signal can be written as the sum of a finite number of slow-varying sinusoids and a small residue. This was formulated by J. McAulay and T. F. Quatieri in 1986 for speech analysis and synthesis [MQ86]. In 1989 X. Serra improved this method by introducing a noise descriptor on the residue so that it doesn't have to be small [Serra97]. The sinusoid model is given as (2.1):

$$x_n = \sum_{m=1}^{M} x_n^m + r_n = \sum_{m=1}^{M} a_n^m \cos\varphi_n^m + r_n$$

where x^m (m=1, 2, …, M) are sinusoids and r is the residue. A complete sinusoid modeling system includes an analyzer and a synthesizer. The analyzer finds the sinusoids and estimates their parameters in (2.1) at a set of measurement points. It contains a sinusoid detector, a parameter estimator, and a partial tracker. Details of these parts have already been reviewed in §2.1~§2.4. The synthesizer simply sums up a finite number of resynthesized sinusoids. Techniques for resynthesizing a sinusoid from incomplete parameter estimates have been reviewed in §2.5. Sinusoid modeling can be used for deterministic component extraction, noise removal, pitch shifting, time stretching, sound morphing, audio coding, etc. However in the literature the use of sinusoid modeling is discussed mostly in an analysis-synthesis framework, where a new sound is generated from the original. These include sound reconstruction with sinusoids (known as spectral modeling synthesis, SMS), time stretching, pitch shifting, frequency warping, etc., which are extensively discussed in [MQ86, Serra97, Zölzer02].

The plain sinusoid model does not explicitly represent individual sound sources. There have been attempts to study pitched sound by grouping individual partials [BC94, Ellis96] that fit in a harmonic relationship. In [Tolonen99] this grouping is applied to the sinusoid model to extract pitched events. After the grouping the sinusoid model (2.1) is converted to a harmonic sinusoid model (1.7b). On the other hand, the partial tracking can also be regarded as a kind of grouping, i.e. the grouping of short-time sinusoid atoms onto partials. In this sense the grouping of sinusoid tracks into harmonic sinusoid tracks is equivalent to the grouping of sinusoid atoms with higher priority on partial continuity than on harmonicity. Since the tracker groups the atoms without considering harmonicity, the harmonic structure is likely to be corrupted at this stage. That is, a valid sinusoid track may fail to fit in a harmonic model, as shown in Figure 2.7a.

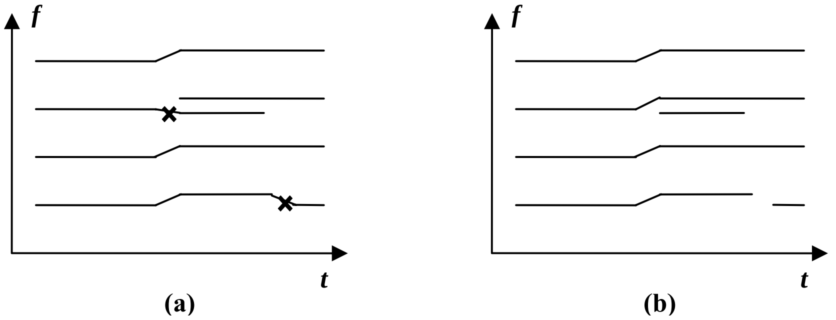


Figure 2. 7 Preserving harmonic structure using harmonicity-priority tracking

(a) continuity-priority tracking; (b) harmonicity-priority tracking

Tracking mistakes are marked with "×" symbols.

Intuitively, the harmonic structure is corrupted when one of the harmonics goes astray onto another under the optimal continuity criterion. This type of error can be alleviated by limiting the duration of each track so that the crossing points are "quarantined" on their own segment without disturbing the other "clean" segments. The short sinusoidal components can then be grouped within their own intervals, and then connected together in time with multi-partial continuity considerations.

If we push the above treatment to the extreme, i.e. each segment is unit-length so that no source-crossing could ever happen, we arrive at an alternative way for obtaining harmonic sinusoids, i.e. grouping harmonic atoms before tracking them over time. By doing this we are giving partial harmonicity higher priority over continuity, so that the tracking never goes out of harmonicity, as shown in Figure 2.7b. This is our starting point of harmonic sinusoid modeling. A harmonic sinusoid analyzer therefore consists of a module for finding harmonic partials at a certain frame, a module for tracking harmonic partials over time, and a module for estimating sinusoidal parameters. We compare the block diagrams of sinusoid and harmonic sinusoid modeling systems in Figure 2.8.

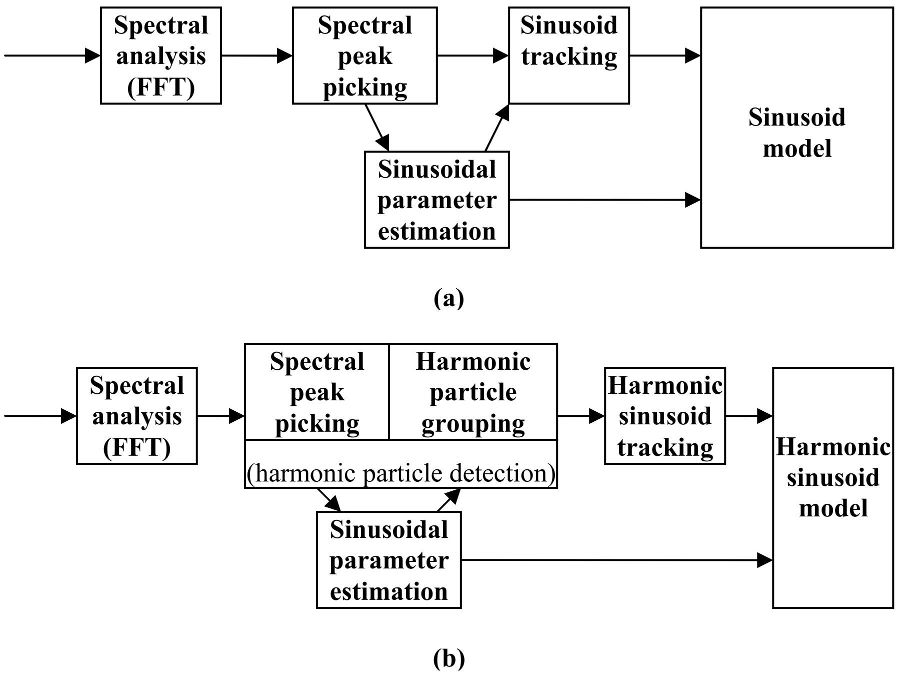


Figure 2. 8 Comparing sinusoid and harmonic sinusoid analysis systems

(a) sinusoid analysis; (b) harmonic sinusoid analysis

While in sinusoid analysis we group sinusoid atoms over time immediately after they are detected, in harmonic sinusoid analysis a harmonic grouping stage is introduced between those two steps to collect sinusoid atoms into harmonic particles, which are then tracked with a harmonic sinusoid tracker. In both systems the tracking modules are based on time continuity structures. Table 2.3 lists the elements of harmonic sinusoid modeling compared to their sinusoid modeling counterparts. The key upgrade from sinusoid to harmonic sinusoid modeling is the replacement of a sinusoid atom with a harmonic particle, which is listed in bold. Accordingly all modules and criteria designed for sinusoid atoms are upgraded for working with harmonic particles.

Sinusoid modeling	Harmonic sinusoid modeling
Object: pure tones	Object: tonal sounds
Sinusoid atoms (peaks)	**Harmonic sinusoid particles**
Spectral peak picking	Harmonic particle detection
Peak amplitude	Amplitude vector, on partial index
Peak frequency	Frequency vector, on partial index
	Inharmonicity
Peak continuity	Harmonic particle continuity
Peak frequency variation	Pitch variation
Peak amplitude variation	Amplitude vector variation
Phase continuity	Multi-phase continuity
	Energy distribution in partials
Peak tracking (sinusoid tracking)	Harmonic sinusoid tracking

Table 2. 3 Comparing sinusoid and harmonic sinusoid modeling

In the table we have used the name *harmonic particle detection* to refer to the combination of peak picking with harmonic grouping. The key element in sinusoid modeling, the sinusoid atom, is detected by peak picking and described by its amplitude, frequency and phase angle. Its harmonic counterpart, the harmonic particle, is detected by harmonic particle detection (see §3.2), and described by amplitude, frequency and phase angle vectors. From the frequency vector we derive inharmonicity (see 3.2.2), describing how partial frequencies depart from perfect harmonicity. The sinusoid tracker tracks sinusoid particles according to peak continuity criteria. The harmonic tracker does its job according to harmonic particle continuity criteria (see §4.1 and §4.2). When partials of pitched sounds are regarded as independent components, the harmonic sinusoid modeling is reduced to sinusoid modeling.

As is the case with sinusoid modeling, the harmonic sinusoid modeling also includes a synthesis module. However, since the conversion from a harmonic sinusoid model to a sinusoid model is trivial, the harmonic sinusoid synthesis can be done in the same way as in sinusoid modeling.

2.7 Summary

In this chapter of technology review we have focused on the main aspects of sinusoid modeling, i.e. sinusoid detection and sinusoidal parameter estimation, partial tracking, and synthesis algorithms. We have also reviewed how these modules are integrated together as a sinusoid modeling system, and how this system is transformed into a harmonic sinusoid modeling system by the introduction of harmonic sinusoid particles. Theoretically a harmonic sinusoid analyzer can be implemented by upgrading the modules of a sinusoid modeling system to work with harmonic particles. This is to be discussed in details in the following chapters.

Chapter 3

Harmonic sinusoid measurements

This chapter is devoted to techniques for estimating the parameters of harmonic particles, i.e. the frequencies, amplitudes and phase angles, from audio signal. A harmonic particle is represented by the number of partials M, partial frequencies f^1, …, f^M, amplitudes a^1, …, a^M, and phase angles φ^1, …, φ^M. To determine a harmonic particle, we locate its partials and estimate the parameters. The harmonic grouping requires the knowledge of amplitudes and frequencies of the atoms. However, good estimation of sinusoidal parameters requires the knowledge of parameter variation, which is only available after the tracking stage (see Chapter 4). Fortunately, parameter accuracy is not critical in the harmonic grouping and tracking stages, which are designed to tolerate parameter errors. We perform parameter estimation in two stages: a pre-tracking estimation that provides rough estimates for use in harmonic grouping and tracking, and a post-tracking estimation that provides the final model parameters.

This chapter is arranged as follows. 3.1 discusses a least-square-error method for estimating stationary sinusoidal parameters in the presence of noise. 3.2 discusses the grouping of sinusoid atoms into harmonic particles. In 3.3 and 3.4 we propose two schemes for post-tracking parameter estimation, which use the knowledge of partial tracks to improve parameter estimation.

3.1 Least-square-error estimation of stationary sinusoids

Now we consider the problem of evaluating parameter estimates. Let x be a sinusoid $x_n=a_n\cos\varphi_n$ with already known parameters, $\hat{a}$, $\hat{f}$ and $\hat{\varphi}$ be its parameter estimates at point 0. It is straightforward to compute the individual errors at time 0:

$$e_a = \hat{a} - a_0,\ e_\varphi = \hat{\varphi} - \varphi_0,\text{ and } e_f = \hat{f} - \frac{\varphi'(0)}{2\pi}. \tag{3. 1}$$

These errors are useful for evaluating analyzer performance on each parameter. However, there are two problems. First, they do not provide an overall performance measure. While we may use these errors to compare systems for each parameter, it is hard to compare analyzers each of which has its own strong points. For instance, if system A is better in frequency estimation than system B, but worse in amplitude estimation, then to compare the two systems we need to know how much frequency error can be "traded" for a certain amount of amplitude error, etc. The second problem of direct parameter comparison is that it requires that we know the instantaneous parameters at the measurement points, which is only applicable to synthesized signals.

To avoid these complications we take a synthesis approach. Given a set of parameters a, f, φ and their estimates $\hat{a}$, $\hat{f}$, $\hat{\varphi}$, we synthesize a sinusoid from each set and compare their waveforms, i.e.

$$e = \sum_n w_n^2 \left(\hat{a}\cos(2\pi\hat{f}n + \hat{\varphi}) - a\cos(2\pi fn + \varphi)\right)^2 \tag{3. 2a}$$

where w is a low-pass window function centred at the measure point, so that the sum is bounded. Comparing two parameter sets with (3.2a) enables each individual parameter error to contribute in its natural way to a global error. When we have no ground truth of the instantaneous parameters, we can use

$$e = \sum_n w_n^2 \left(\hat{a}\cos(2\pi\hat{f}n + \hat{\varphi}) - x_n\right)^2 \tag{3. 2b}$$

The above time-domain formulation has been given in previous literatures e.g. [Kay87]. The frequency-domain version of (3. 2b) is

$$e = \frac{1}{N}\sum_{k=0}^{N-1}\left|\frac{\hat{a}}{2}e^{j\hat{\varphi}}W\left(k/N - \hat{f}\right) + \frac{\hat{a}}{2}e^{-j\hat{\varphi}}W\left(k/N + \hat{f}\right) - X_k\right|^2 \tag{3. 3a}$$

where N is the size of w, W is the DTFT of w, and X is the windowed DFT of x. Most energy of W is concentrated within a narrow band, say 2B bins, centred at 0. Then using the symmetry of Fourier transforms of real signals, for the typical frequency range we work in, we may write

$$e \cong \frac{1}{N}\left(\left|X_0\right|^2 + \left|X_{\frac{N}{2}}\right|^2 + 2\sum_{\substack{1\le k\le N/2-1 \\ k\notin(N\hat{f}-\mathrm{B},N\hat{f}+\mathrm{B})}} \left|X_k\right|^2 + 2\sum_{N\hat{f}-\mathrm{B}<k<N\hat{f}+\mathrm{B}} \left| \frac{\hat{a}}{2}e^{j\hat{\varphi}}W\left(\frac{k}{N}-\hat{f}\right)-X_k \right|^2 \right) \qquad (3.\ 3b)$$

"$\cong$" is used in (3.3b) since the energy of W outside the 2B bins is ignored. The part of e that concerns parameter estimates is the band-limited square error:

$$e^B = \frac{1}{N}\sum_{N\hat{f}-\mathrm{B}<k<N\hat{f}+\mathrm{B}} \left| \frac{\hat{a}}{2}e^{j\hat{\varphi}}W\left(k/N-\hat{f}\right)-X_k^w \right|^2 \qquad (3.\ 3c)$$

(3.3c) separates from (3.3a) a narrow band that contains most energy of the sinusoid. Since W itself is band-limited, using e^B instead of e hardly affects the estimation outcome, but effectively reduces the frequency-domain computation involved in calculating the error and its derivatives.

Given two analyzers, we may compare them using (3.3c) for an arbitrary signal. The one with a lower e^B is a better estimator in the sense of less square error. We define a least-square-error (LSE) estimator as the one that minimizes e^B in (3.3c).

It can be shown that the LSE estimator is equivalent to the Gaussian maximal likelihood (ML) estimator [Kay87].

3.1.1 The estimating process

Given a windowed DFT X_k, the LSE estimator finds a set of parameter estimates $\hat{a}$, $\hat{f}$ and $\hat{\varphi}$ that minimizes e^B. We always start from a rough estimate of the frequency, say $\hat{f}_0$. This can be obtained by any method discussed in §2.2. We choose k_1 as the minimal integer above $N\hat{f}_0-\mathrm{B}$, k_2 as the maximal integer below $N\hat{f}_0+\mathrm{B}$, then redefine the band-limited square error as

$$e^B\left(\hat{a},\hat{f},\hat{\varphi}\right) = \frac{1}{N}\sum_{k=k_1}^{k_2} \left| \frac{\hat{a}}{2}e^{j\hat{\varphi}}W\left(k/N-\hat{f}\right)-X_k^w \right|^2 \qquad (3.\ 4a)$$

By fixing the summation bounds instead of letting them vary with $\hat{f}$, we avoid possible discontinuity problems of e^B at bin boundaries. We further introduce the scalar $\lambda=ae^{j\varphi}/2$, as well as the vectors

$$\mathbf{W}(f)=\begin{bmatrix} W(k_1/N-f) \\ W((k_1+1)/N-f) \\ \vdots \\ W(k_2/N-f) \end{bmatrix},\ \mathbf{X}=\begin{bmatrix} X_{k_1} \\ X_{k_1+1} \\ \vdots \\ X_{k_2} \end{bmatrix}, \tag{3. 4b}$$

so

$$e^B(\lambda,f)=\frac{1}{N}\|\lambda\mathbf{W}(f)-\mathbf{X}\|^2 \tag{3. 4c}$$

The optimization of e^B regarding $\{\hat{a},\hat{f},\hat{\varphi}\}$ does not have a closed-form solution for an arbitrary window function, so we turn to numerical searching methods. Fortunately, if $\hat{f}$ is fixed, we do have an analytical solution. This helps to reduce the searching scope from three dimensions to one.

We consider the optimization of $\{\hat{a},\hat{\varphi}\}$, or $\hat{\lambda}=\hat{a}e^{j\hat{\varphi}}/2$, with fixed $\hat{f}$. The solution is straightforward:

$$\hat{\lambda}(\hat{f})=\frac{<\mathbf{X},\mathbf{W}(\hat{f})>}{\|\mathbf{W}(\hat{f})\|^2} \tag{3. 5a}$$

and the square error at this $\hat{\lambda}$ is

$$e^B(\hat{f})=\frac{1}{N}\left(\|\mathbf{X}\|^2-|\hat{\lambda}(\hat{f})|^2\|\mathbf{W}(\hat{f})\|^2\right)=\frac{1}{N}\left(\|\mathbf{X}\|^2-\frac{|<\mathbf{X},\mathbf{W}(\hat{f})>|^2}{\|\mathbf{W}(\hat{f})\|^2}\right) \tag{3. 5b}$$

$\hat{\lambda}(f)$ is interpreted as the orthogonal projection of $\mathbf{X}$ onto the 1-D subspace containing $\mathbf{W}(f)$. (3.5b) expresses the least square error as a function of $\hat{f}$ only. Once we have optimized e^B regarding variable $\hat{f}$ using (3.5b), the optimal $\hat{a}$ and $\hat{\varphi}$ can be calculated immediately with (3.5a). The optimal $\hat{f}$ can be found using a standard one-dimension optimization method. If W has an analytic expression, we can use the Newton method. If not, we can run a convex 1-D search.

To use the Newton method we need to calculate the 1st- and 2nd-order derivatives of $e^B(f)$ regarding f. For the cosine window family, we formulate the details of the computations in Appendix C.1 and C.2.

We summarize the LSE estimator as follows.

$$\boxed{\begin{gathered} \hat{f} = \arg\max_f \frac{|<\mathbf{X}, \mathbf{W}(f)>|^2}{\|\mathbf{W}(f)\|^2} \\ \hat{\lambda} = \frac{<\mathbf{X}, \mathbf{W}(\hat{f})>}{\|\mathbf{W}(\hat{f})\|^2} \\ \hat{a} = 2|\hat{\lambda}| \\ \hat{\varphi} = \arg\hat{\lambda} \end{gathered}} \qquad (*)$$

Other formulations equivalent to (*) are found in e.g. [Kay87] with rectangular windowing, and in [Virtanen00] with general windowing. Multi-sinusoid version of (*) is also discussed in [Kay87] and used for sinusoid modeling in [Tolonen99, Virtanen00].

The variation of the denominator $\|\mathbf{W}(\hat{f})\|^2$ with $\hat{f}$ is an effect of the band-limiting, and hardly has any substantial importance. In a simplified version we take an approximation by ignoring the marginal energy outside the considered band, and write

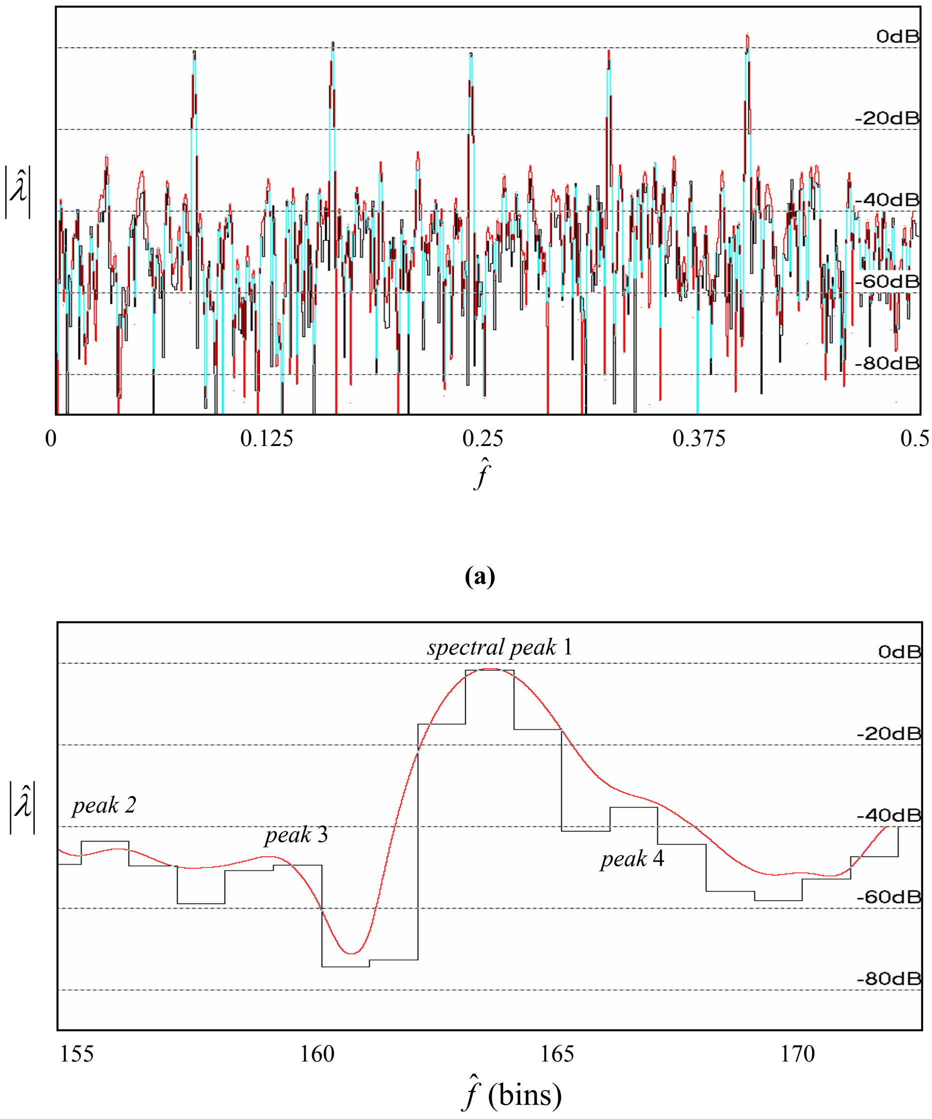
$$\|\mathbf{W}(\hat{f})\|^2 = \|\mathbf{W}(0)\|^2 = N\sum_n w_n^2, \qquad (3.\ 6)$$

This leads to the maximal-cross-correlation method [Rodet97]:

$$\hat{f} = \arg\max_f |<\mathbf{X}, \mathbf{W}(f)>|^2,\ \hat{\lambda} = \frac{<\mathbf{X}, \mathbf{W}(\hat{f})>}{\|\mathbf{W}(0)\|^2} \qquad (3.\ 7)$$

However, by using (3.7) we are deliberately casting away side-lobe energies of w, which may introduce a bias for some window functions.

Figure 3.1 depicts $|\hat{\lambda}(\hat{f})|$ as a function of $\hat{f}$. The signal is a five-partial harmonic sinusoid with fundamental frequency 0.08 and partial amplitudes 1, plus white noise with SNR=0dB. A Hann window of size 1024 is used. The DFT is given in black and $\left|\hat{\lambda}(\hat{f})\right|$ in red; where the two overlap it appears in aqua. Figure 3.1 (a) shows the complete Nyquist frequency range, while Figure 3.1 (b) zooms in near the second partial.



(a)

(b)

Figure 3. 1 Projection amplitude as a function of frequency

(a) the whole Nyquist range; (b) around a local spectral peak

3.1.2 Multiple spectra

Some musical instruments have constant pitches. In this case it is reasonable to evaluate a constant frequency from multiple short-time spectra. Let the band-limited spectrum vector of the l^{th} frame be $\mathbf{X}_l$, the amplitude-phase factor estimate of the l^{th} frame be $\hat{\lambda}_l$, the frequency estimate be $\hat{f}$, then we define the total windowed square error as

$$e^B\left(\hat{\lambda}_1, \hat{\lambda}_2, \cdots, \hat{\lambda}_L, \hat{f}\right) = \frac{1}{N}\sum_l \left\|\hat{\lambda}_l \mathbf{W}(\hat{f}) - \mathbf{X}_l\right\|^2 \tag{3. 8a}$$

For a fixed $\hat{f}$ the above is minimized by setting

$$\hat{\lambda}_l(\hat{f}) = \frac{<\mathbf{X}_l, \mathbf{W}(\hat{f})>}{\left\|\mathbf{W}(\hat{f})\right\|^2}, \ \forall\ l \tag{3. 8b}$$

The minimal square error fixing $\hat{f}$ is:

$$e^B\left(\hat{f}\right) = \frac{1}{N}\sum_l \left\|\mathbf{X}_l\right\|^2 - \frac{1}{N}\sum_l \left|<\mathbf{X}_l, \mathbf{W}(\hat{f})>\right|^2 \left\|\mathbf{W}(\hat{f})\right\|^{-2} \tag{3. 8c}$$

To minimize this error we search for an optimal frequency:

$$\hat{f} = \arg\max_f \frac{\sum_l \left|<\mathbf{X}_l, \mathbf{W}(f)>\right|^2}{\left\|\mathbf{W}(f)\right\|^2} \tag{3. 8d}$$

3.1.3 Non-stationary sinusoids

Since the LSE method is based on the orthogonal projection of the signal spectrum onto the subspace of fixed-frequency stationary sinusoid spectra, it does not work for non-stationary sinusoids in the same way as for stationary ones. To interpret the LSE frequency estimate, we apply the LSE estimator, in its simplified version, to time-varying sinusoids and get the following result.

Proposition 3.1 Let x be a complex sinusoid given as

$$x_n = a_n e^{j(\varphi_c + 2\pi\int_{N/2}^{n} f(t)dt)}, \tag{3. 9a}$$

w be a window function, and $\hat{f}$ be the LSE frequency estimation of x, then

$$\hat{f}=\frac{\sum_{n=0}^{N-1}\sum_{m=0}^{N-1}w_{mn}a_n a_m\int_m^n f(t)dt\,\mathrm{sinc}\frac{\Delta\varphi_{mn}}{\pi}}{\sum_{n=0}^{N-1}\sum_{m=0}^{N-1}w_{mn}a_n a_m(n-m)\,\mathrm{sinc}\frac{\Delta\varphi_{mn}}{\pi}}, \tag{3. 9b}$$

where $\Delta\varphi_{mn}=2\pi\left(\int_m^n f(t)dt-(n-m)\hat{f}\right)$, w_{mn}=$(n$-$m)w_m^{\ 2}w_n^{\ 2}$.

The proof is given in Appendix C.3. By rearranging the summing indices of (3.9b) we get

$$\hat{f}=\frac{\sum_{l=0}^{N-2}\eta_l\int_0^1 f(l+t)dt}{\sum_{l=0}^{N-2}\eta_l},\ \eta_l=\sum_{n=l+1}^{N-1}\sum_{m=0}^{l}w_{mn}a_n a_m\,\mathrm{sinc}\frac{\Delta\varphi_{mn}}{\pi}. \tag{3. 9c}$$

(3.9c) says that the LSE frequency estimate is a weighted average of the instantaneous frequency during the interval from which the spectrum is calculated. This result is similar to (2.7c) derived for the standard FFT method, except for the definition of w_{mn}. Here it is the squares of w_m and w_n, rather than are w_m and w_n themselves, that produce w_{mn}, which further favours the central part of the window.

Strictly speaking, when the frame involves large frequency variation, η_l is not guaranteed to be non-negative when $f(l)$ departs far from $\hat{f}$. If the frequency varies in a smooth way, the sign of η_l appear alternatively positive and negative with l during the time $f(l)$ remains far from $\hat{f}$. The effect is that they partially cancel the contribution of each other in (3.9c).

The amplitude-phase factor $\hat{\lambda}$ can still be interpreted as the maximal projection of x onto any subspace of fixed-frequency sinusoids. However, since time-varying sinusoids can not be effectively approximated in such a subspace, this projection can no longer be used for amplitude estimation. The parameter estimation of time-varying sinusoids will be discussed in §3.3.

It is easily shown by (3.9c) that the LSE frequency estimate is accurate for linear chirps, i.e. $\hat{f}$ equals the instantaneous frequency at the frame centre, if x has constant amplitude and linear frequency. A closer study shows that if the frequency is odd-symmetric (allowing a constant shift) and the amplitude is even-symmetric within the analysis window, then the LSE estimator does not incur a frequency estimation error. A discussion of parameter symmetry and its effect on (3.9c) is included in Appendix C.3.

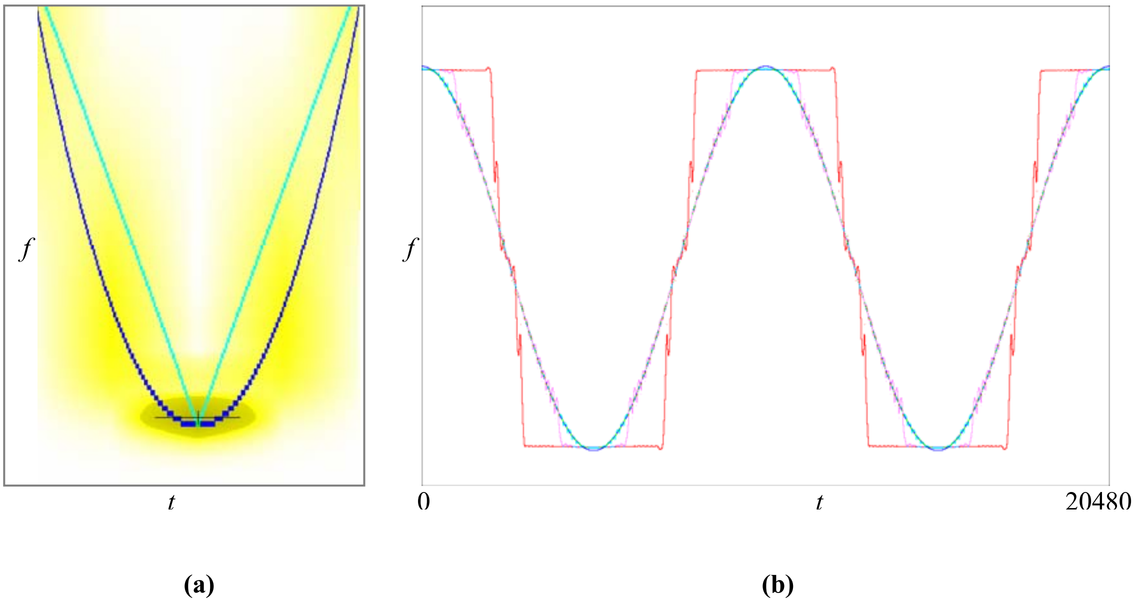


Figure 3. 2 Frequency averaging effect in the LSE method

(a) at a peak of a frequency track; (b) on a whole frequency track modulated by a sinusoid

Figure 3.2 shows the frequency averaging effect in the LSE method. Figure 3.2 (a) shows a quadratic frequency track (dark blue). The gray-scale shadow behind the frequency track is the spectrogram. When we try to estimate the instantaneous frequency at the bottom of the track using the spectrum, the frequency estimate is obtained at the position marked with a cross, which is a local average of the instantaneous frequency track. In Figure 3.2 (b) we measure the instantaneous frequency of a frequency-modulated sinusoid with different frame widths. The modulator period is 10240 samples. An instantaneous frequency is estimated for each point in time at the global spectral peak calculated from a frame centred at that point. Frequency tracks estimated using 2048-, 4096- and 8192-point windows are given in aqua, pink and red respectively. Large estimation errors are observed near the peaks of frequency modulation, since these positions have the largest even-symmetric components. The comparison between window sizes suggests the use of short windows in the presence of large signal dynamics. This issue will be discussed in §3.4.

3.2 Detecting harmonic particles

Harmonic sinusoid particles are detected in a single frame of data. Given the fundamental frequency f^1, we expect to find sinusoidal peaks at multiples of f^1 on the frequency axis. Theoretically, a harmonic partial appears as a spectral peak, with its frequency being a multiple of that of a fundamental partial. Based on such an assumption it seems plausible that a harmonic particle can be found by grouping short-time sinusoid atoms, whose frequencies have been estimated with the LSE or some other method, that are multiples of a fundamental. However, there are two matters to consider: that the frequencies are all estimates, each carrying an error, and that there may be inharmonicity among the partial frequencies. We address them separately.

3.2.1 Frequency estimation error

The frequency estimate can be very accurate when the pitch is stable and the partial is free of noise or disturbance. In real-world recordings the pitch line can have smooth (glissando) or repetitive (vibrato) variations fast enough to affect frequency estimates, and noise and concurrent sinusoids do disturb sinusoid analyzers. Frequency variation, and accompanying amplitude variation, shifts the frequency estimate from the instantaneous frequency at the window centre to another weighted frequency centroid as shown in (3.9c). The existence of noise or nearby sinusoids further shifts this peak in even less predictable manner. Therefore we should always allow an error for a frequency estimate, i.e. we bound the true frequency f^m by

$$\hat{f}^m - \hat{\delta}^m < f^m < \hat{f}^m + \hat{\delta}^m \tag{3.10}$$

where $\hat{\delta}^m$ is the upper bound of the error.

In the worst case, the partial of interest may not stand out as a spectral peak at all. This happens in two cases: either it is too weak itself, or it is shaded by noise or nearby stronger partials. There are multiple reasons for weak partials. Some instruments, such as the pipe organ and the clarinet, leave out certain partials by design. On some instruments designed to present a full collection of harmonic partials, the player has the choice to suppress certain partials by his playing technique. The majority of weak partials happen due to the filtering effect of the instrument. When formants are present, for example, partials that happen to be located at a node become weak partials. This effect is frequently encountered by instruments with variable pitches: as the frequency tracks vary freely, one observes a dependency of amplitude on frequency. Partials may also become weak due to beating with a close sinusoid, which is typical of the grand piano. In harmonic sinusoid modeling we say a partial is weak if it does not appear as a spectral peak and there is no audible signal power in a narrow band in which it is expected, and a partial is masked if it does not appear as a spectral peak but there is audible signal power in a narrow band in which it is expected. In either case there is little hope to estimate an accurate partial frequency from the spectrum.

3.2.2 Inharmonicity

Inharmonicity is the phenomenon that harmonic partial frequencies depart from multiples of the fundamental. Real-world free vibrating bodies more or less depart from perfect harmonicity. The most encountered example is the inharmonicity of a stiff string. [Klapuri99] gives an example of explicitly expressing the partial frequencies with the fundamental frequency f^1 and a stiffness coefficient B:

$$f^m(f^1, B) = mf^1 \cdot [1 + B(m^2 - 1)]^{1/2} \tag{3. 11a}$$

The departure of the m^{th} partial from the perfect harmonic frequency is

$$\delta f^m = mf^1 \frac{B(m^2 - 1)}{[1 + B(m^2 - 1)]^{1/2} + 1} \tag{3. 11b}$$

B is a constant for a given string, typically well below 0.001.

In [Klapuri99] an exhaustive search is suggested for finding the coefficient B, based on the observation that it does not significantly increase the overall computation cost including spectral analysis. [RLV07] proposes an iterative method for finding B more efficiently, which relies on relatively accurate estimates of partial frequencies, especially the fundamental. [GD05] implements (3.11a) within a Bayesian harmonic model, where B is introduced as a model parameter to be estimated. In the following we implement the inharmonicity model within an inequality-based harmonic partial finder, which jointly "brackets" f^1 and B within a small range.

According to (3.11b), the frequency departure is positive, proportional to f^1, and roughly proportional to m^3 when m is not very large. The true frequency of the m^{th} partial, however, may still have an error from (3.11a) for two reasons: that (3.11a) itself comes with some mathematical approximation [FR98], and that there are still mechanisms other than stiffness that can shift partial frequencies. When the stiffness is the main source of inharmonicity, this additional error is usually very small compared to δf^m.

To generalize, we assume that there is only one main mechanism of inharmonicity, associated with a coefficient B, so that the m^{th} partial frequency is approximately shifted to $f^m(f^1,B)$, $f^m(f^1,0)=mf^1$. The true partial frequency f^m is then bounded by

$$f^m(f^1,B)-\delta^m < f^m < f^m(f^1,B)+\delta^m, \tag{3. 12}$$

where δ^m represents the error involved in $f^m(f^1,B)$ itself , and frequency shifts caused by all the less-important mechanisms.

3.2.3 Modeling harmonic partial frequencies with inequality system

Now we relate the partial frequency model to the frequency estimates. For each partial, we have three frequencies: the true frequency f^m, the model frequency $f^m(f^1,B)$, and the estimate $\hat{f}^m$. The true frequency and the model frequency are related by (3.12). The departure of the estimate $\hat{f}^m$ from the true frequency is described by (3.10). Combining the two we get

$$f^m(f^1,B)-\delta^m-\hat{\delta}^m < \hat{f}^m < f^m(f^1,B)+\delta^m+\hat{\delta}^m \tag{3. 13a}$$

There is no hint for evaluating δ^m except it is very small. However, since $\hat{\delta}^m$ is also a rough estimate, it is reasonable that we combine the two error bounds as one, either by ignoring δ^m, or by expanding $\hat{\delta}^m$ by a small amount. We denote this combined error bound Δ^m:

$$f^m(f^1,B)-\Delta^m<\hat{f}^m<f^m(f^1,B)+\Delta^m \tag{3. 13b}$$

(3.13b) relates the model parameters f^1 and B to the observations $\hat{f}^1$, $\hat{f}^2$, $\hat{f}^3$, …. Both the model frequency $f^m(f^1,B)$ and the true frequency f^m are invisible to us (hidden). However, they are indirectly observed through the frequency estimates. To find a harmonic particle from the spectrum, we look through the spectral peaks for those that satisfy (3.13b) for some reasonable f^1 and B. Regarding harmonic grouping we raise three questions:

1. given the frequency estimates $\hat{f}^{m_1}$, $\hat{f}^{m_2}$, …, $\hat{f}^{m_M}$ of the m_1th, m_2th, …, m_Mth partial, can they be grouped as harmonic partials?

2. given a group of harmonic partials $\hat{f}^{m_1}$, $\hat{f}^{m_2}$, …, $\hat{f}^{m_M}$, can another frequency $\hat{f}^{m_{M+1}}$, with the partial index m_{M+1}, be associated to the group at another partial?

3. given a group of harmonic partials $\hat{f}^{m_1}$, $\hat{f}^{m_2}$, …, $\hat{f}^{m_M}$, how do we estimate the fundamental f^1 and inharmonicity coefficient B?

We give answers to these questions in the following sub-sections.

3.2.4 Finding the f^1-B range from partial frequencies

Given the frequency estimates $\hat{f}^{m_1}$, $\hat{f}^{m_2}$, …, $\hat{f}^{m_M}$ of the m_1th, m_2th, …, m_Mth partial, we have the following inequalities:

$$\begin{cases} f^{m_1}(f^1,B)-\Delta^{m_1}<\hat{f}^{m_1}<f^{m_1}(f^1,B)+\Delta^{m_1} \\ f^{m_2}(f^1,B)-\Delta^{m_2}<\hat{f}^{m_2}<f^{m_2}(f^1,B)+\Delta^{m_2} \\ \qquad\qquad\qquad \cdots\cdots \\ f^{m_M}(f^1,B)-\Delta^{m_M}<\hat{f}^{m_M}<f^{m_M}(f^1,B)+\Delta^{m_M} \end{cases} \tag{3. 14a}$$

This is a system of inequalities of independent variables f^1 and B. Let the solution set of this inequality system be R. The group of frequencies can be regarded as harmonic partial frequencies if and only if R≠Φ.

For the stiff string model (3.14a) becomes

$$\begin{cases} m_1 f^1\sqrt{1+B(m_1^2-1)}-\Delta^{m_1} < \hat{f}^{m_1} < m_1 f^1\sqrt{1+B(m_1^2-1)}+\Delta^{m_1} \\ m_2 f^1\sqrt{1+B(m_2^2-1)}-\Delta^{m_2} < \hat{f}^{m_2} < m_2 f^1\sqrt{1+B(m_2^2-1)}+\Delta^{m_2} \\ \qquad\qquad \cdots\cdots \\ m_M f^1\sqrt{1+B(m_M^2-1)}-\Delta^{m_M} < \hat{f}^{m_M} < m_M f^1\sqrt{1+B(m_M^2-1)}+\Delta^{m_M} \end{cases} \tag{3. 14b}$$

(3.14b) is highly nonlinear in the f^1-B space, which makes R hard to represent. We linearize (3.14b) by observing that

$$mf^1\sqrt{1+B(m^2-1)}-\Delta^m < \hat{f}^m < mf^1\sqrt{1+B(m^2-1)}+\Delta^m \tag{3.15a}$$

is equivalent to

$$\left(\frac{\hat{f}^m-\Delta^m}{m}\right)^2 < \left(f^1\right)^2+\left(f^1\right)^2 B(m^2-1) < \left(\frac{\hat{f}^m+\Delta^m}{m}\right)^2 \tag{3. 15b}$$

Using the variable substitution

$$F=(f^1)^2,\ G=FB=B(f^1)^2, \tag{3. 16}$$

(3.14b) becomes

$$\begin{cases}\left(\dfrac{\hat{f}^{m_1}-\Delta^{m_1}}{m_1}\right)^2 < F+(m_1^2-1)G < \left(\dfrac{\hat{f}^{m_1}+\Delta^{m_1}}{m_1}\right)^2 \\ \left(\dfrac{\hat{f}^{m_2}-\Delta^{m_2}}{m_2}\right)^2 < F+(m_2^2-1)G < \left(\dfrac{\hat{f}^{m_2}+\Delta^{m_2}}{m_2}\right)^2 \\ \qquad\qquad\cdots\cdots \\ \left(\dfrac{\hat{f}^{m_M}-\Delta^{m_M}}{m_M}\right)^2 < F+(m_M^2-1)G < \left(\dfrac{\hat{f}^{m_M}+\Delta^{m_M}}{m_M}\right)^2 \end{cases} \tag{3. 17}$$

(3.17) is a linear inequality system. Its solution set is the f^1-B range R mapped into the F-G space by (3.16). The inverse mapping is

$$f^1=\sqrt{F}\,,\ B=G/F \tag{3. 18}$$

The frequencies $\hat{f}^{m_1}$, $\hat{f}^{m_2}$, …, $\hat{f}^{m_M}$ can be regarded as harmonic partial frequencies if and only if the solution set R of (3.17) is non-empty. In the F-G plane R is a convex polygon. Let the number of its vertices be N and the n^{th} vertex be (F_n, G_n), $0\leq n<N$, then R can be represented using the vertices by $\{N;\ (F_n, G_n)_{0\leq n<N}\}$. Details of solving (3.17) are discussed in Appendix D.1.

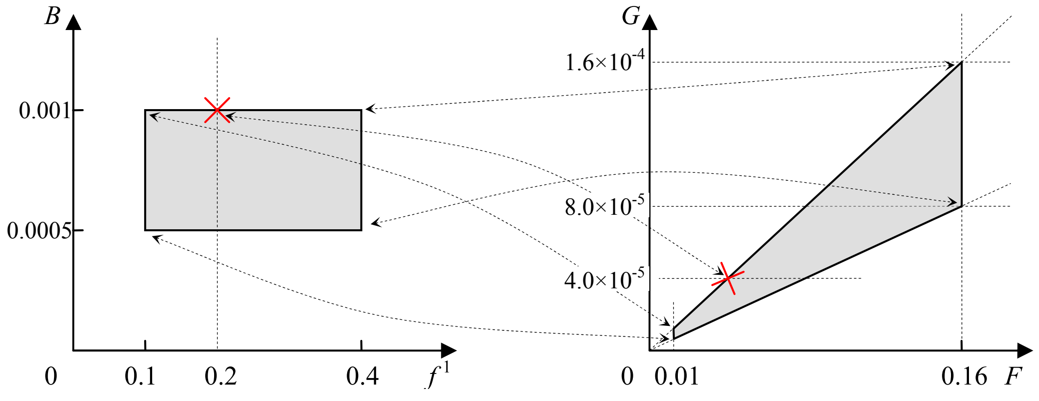


Figure 3. 3 Mapping between the f^1-B and F-G planes

Figure 3.3 shows the mapping between the two planes. A line parallel to the f^1 axis in the f^1-*B* plane is mapped as a line passing through the origin in the *F*-*G* plane; a line parallel to the *B* axis in the f^1-*B* plane is mapped as a line parallel to the *G* axis in the *F*-*G* plane.

R is a representation of the frequency contents of a harmonic particle. For any pair of parameters (f^1, *B*) that falls in R, and only for these pairs, the frequency estimates $\hat{f}^{m_1}$, $\hat{f}^{m_2}$, …, $\hat{f}^{m_M}$ are within the preset error bounds Δ^{m_1}, Δ^{m_2}, …, Δ^{m_M} from the model frequencies.

3.2.5 Adding new frequencies to a harmonic particle

Let R be the range derived from harmonic partials $\hat{f}^{m_1}$, $\hat{f}^{m_2}$, …, $\hat{f}^{m_M}$. Now we have a frequency $\hat{f}^{m_{M+1}}$, associated with partial index m_{M+1}. This new frequency estimate introduces two constraints according to (3.15b). The new frequency is acceptable as a new harmonic partial if R, subject to the two new constraints, does not reduce to Φ.

Another way to determine if $\hat{f}^{m_{M+1}}$ is compatible with the previous partials is to calculate a compatible frequency range and see if $\hat{f}^{m_{M+1}}$ falls in this range. This is especially effective if there are multiple candidates of $\hat{f}^{m_{M+1}}$ to evaluate. Given R, the upper and lower bounds of $f^{m_{M+1}}$ is given by

$$\inf_{(f_1,B)\in R} f^{m_{M+1}}(f_1,B) \le f^{m_{M+1}} \le \sup_{(f_1,B)\in R} f^{m_{M+1}}(f_1,B) \qquad (3.\ 19a)$$

Then the range for the estimate is

$$\inf_{(f_1,B)\in R} f^{m_{M+1}}(f_1,B) - \Delta^{m_{M+1}} \le \hat{f}^{m_{M+1}} \le \sup_{(f_1,B)\in R} f^{m_{M+1}}(f_1,B) + \Delta^{m_{M+1}} \qquad (3.\ 19b)$$

Any frequency estimate that falls in this range is compatible with R, i.e. there is some point (f^1, *B*)∈R so that $\hat{f}^{m_{M+1}}$ is within the preset error bound $\Delta^{m_{M+1}}$ from the model frequency calculated from (f^1, *B*).

In case of the stiff string model, since the square of $f^{m_{M+1}}$ is a linear function of *F* and *G*, and R is a convex polygon in the *F*-*G* plane, the extrema of $f^{m_{M+1}}$ are reached at the vertices. So we have

$$f_{-}^{m_{M+1}}(\mathrm{R}) < f^{m_{M+1}} < f_{+}^{m_{M+1}}(\mathrm{R}) \qquad (3.\ 20\mathrm{a})$$

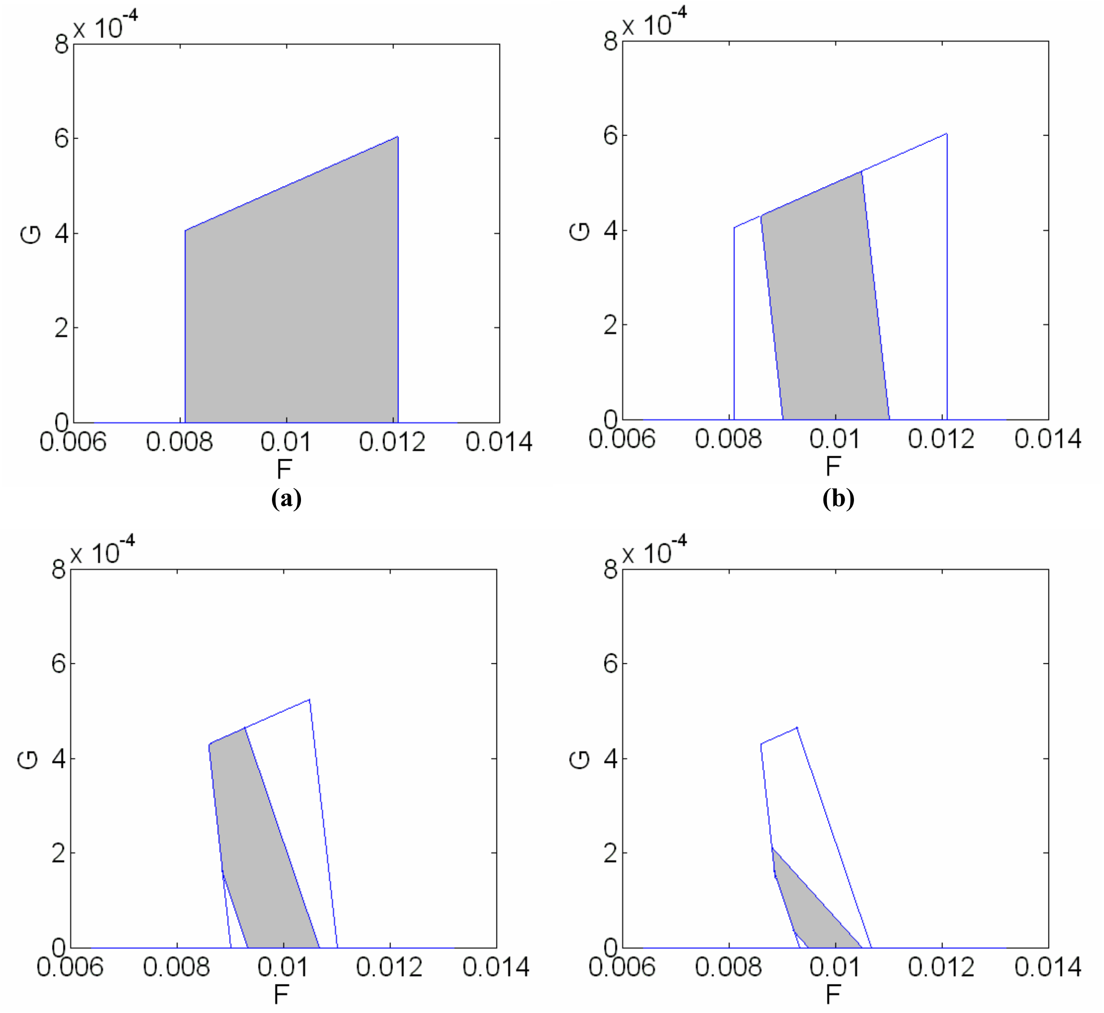
and

$$f_{-}^{m_{M+1}}(\mathrm{R}) - \Delta^{m_{M+1}} < \hat{f}^{m_{M+1}} < f_{+}^{m_{M+1}}(\mathrm{R}) + \Delta^{m_{M+1}} \qquad (3.\ 20\mathrm{b})$$

where

$$f_{-}^{m}(\mathrm{R}) = m\sqrt{\min_{n}(F_n + (m^2-1)G_n)}\,,\ f_{+}^{m}(\mathrm{R}) = m\sqrt{\max_{n}(F_n + (m^2-1)G_n)} \qquad (3.\ 21)$$

If $\hat{f}^{m_{M+1}}$ is compatible with R, adding this new partial updates R by two linear constraints. Figure 3.4 shows how R is updated for a perfect harmonic particle with neither frequency estimation error nor spurious peaks. In this example f^1=0.1 and Δ^m =0.01, 0≤*B*≤0.05. The partials are added in the order of partial index, from 1 to 4. In Figure 3.4 (a) R is initialized using the fundamental partial f^1=0.1 and whole *B* range 0~0.05. R obtained by using the lowest 2, 3, 4 partials are shown in (b), (c), (d) respectively. Each newly added partial "chops off" the part of R outside a band specified by the pair of inequalities associated with the partial. The more partials we use, the smaller R becomes. Details of the operation are given in Appendix D.1.



(a)

(b)

(c)

(d)

Figure 3. 4 Update R using found partials

(a) initial R; (b)(c)(d) updating R using the 1st, 2nd and 3rd partials

3.2.6 Estimating model parameters

The word “model” in the sub-section title refers to the harmonic frequency model with parameters f^1 and B. As mentioned above, R represents the knowledge accumulated from the given partial frequencies as a range in the *F*-*G* plane, or equivalently, in the f^1-B plane. We can derive from R the upper and lower bounds of the model parameters:

$$\inf_{(f^1,B)\in R} f^1 < f^1 < \sup_{(f^1,B)\in R} f^1, \quad \inf_{(f^1,B)\in R} B < B < \sup_{(f^1,B)\in R} B \tag{3. 22}$$

In the stiff string model it is

$$\sqrt{\min_n F_n} < f^1 < \sqrt{\max_n F_n} \tag{3. 23a}$$

$$\min_n \frac{G_n}{F_n} < B < \max_n \frac{G_n}{F_n} \tag{3. 23b}$$

In this model the span of R on the F axis determines the precision of f^1, and the sweep angle of R, with respect to the origin (0, 0), determines the precision of *B*.

However, unless R has zero size (but not empty), it does not provide estimates of f^1 or *B* in a precise form. Accordingly, to estimate the fundamental frequency or the stiffness coefficient, we need to "shrink" R to zero size. To do this we reconsider the allowed error bounds Δ^m, *m*=1, 2, …, by which we derive the range R.

Generally speaking, the choice of Δ^m is a matter of trade-off. Large error bounds improve the robustness against unpredictable frequency estimate errors, but at the same time increase the chance of collecting spurious spectral peaks. Smaller error bounds, on the other hand, may fail to catch the correct partial and make R vanish before enough partials are collected. Since it is important to collect the partials in the first place, we prefer large error bounds in the partial grouping stage. When we need estimates of the fundamental frequency or stiffness coefficient, we may shrink R by reducing the error bounds simultaneously. Let θ be a number between 0 and 1. and the error bound associated with the *m*th partial be $\theta\Delta^m$. For each value of θ we can derive a range R(θ), using $\theta\Delta^m$ as the error bound associated with $\hat{f}^m$, *m*=1. 2, …. When θ equals 1 we get the original R discussed in 3.2.4 and 3.2.5. The size of R(θ) is a monotonic function of θ, and the size of R(0) is 0. So there exists a maximal θ between 0 and 1, say η, so that the size of R(η) is 0, and $\forall\,\theta>\eta$, the size of R(θ) is positive. Using R(η) we can get a more precise range for f^1 and *B* than what R(1) provides in (3.22). In particular, if R is shrunk to a single point at η, which is the usual case, we have estimates of f^1 and *B* in the precise form.

Now the question is, where is η? We consider applying the *M* constraints with the argument θ on R. Given a point (f^1, *B*)$\in$R, it lies on R(θ) if and only if it satisfies all the *M* constraints, i.e.

$$f^m(f^1,B)-\theta\Delta^m < \hat{f}^m < f^m(f^1,B)+\theta\Delta^m\text{, } m=m_1, m_2, m_3, \ldots, m_M. \tag{3. 24a}$$

or

$$\frac{\left|\hat{f}^m - f^m(f^1,B)\right|}{\Delta^m} < \theta\text{, } m=m_1, m_2, m_3, \ldots, m_M. \tag{3. 24b}$$

or

$$\theta > \max_m \frac{\left|\hat{f}^m - f^m(f^1,B)\right|}{\Delta^m} \equiv \theta(f^1,B) \tag{3. 24c}$$

$\theta(f^1, B)$ is the minimal value of θ for R(θ) to cover the point (f^1, B). We define

$$\theta(R) = \inf_{(f^1,B)\in R} \theta(f^1,B) \tag{3. 25}$$

$\theta(R)$ has the following properties:

1) for any $\theta<\theta$(R), R(θ) is empty;

2) for any $\theta>\theta$(R), R(θ) is non-empty.

Therefore we have found $\eta=\theta$(R), or

$$\eta = \inf_{(f^1,B)\in R} \max_m \frac{\left|\hat{f}^m - f^m(f^1,B)\right|}{\Delta^m} \tag{3. 26a}$$

This is a minimal-maximum problem. The model parameters can be estimated at the minimum:

$$(\hat{f}^1,\hat{B}) = \arg \inf_{(f^1,B)\in R} \max_m \frac{\left|\hat{f}^m - f^m(f^1,B)\right|}{\Delta^m} \tag{3. 26b}$$

For the stiff string model this becomes

$$(\hat{F},\hat{G}) = \arg \inf_{(F,G)\in R} \max_m \frac{\left|\hat{f}^m - m\sqrt{F+(m^2-1)G}\right|}{\Delta^m}\text{, } \hat{f}^1 = \sqrt{\hat{F}}\text{, } \hat{B} = \hat{G}/\hat{F} \tag{3. 26c}$$

We call $\frac{\left|\hat{f}^m - f^m(f^1,B)\right|}{\Delta^m}$ the relative frequency estimation error. Equations (3.26a)~(3.26c) show that by shrinking R to zero, we locate the parameter pair that minimizes the maximal relative frequency estimation error of all the given frequency estimates.

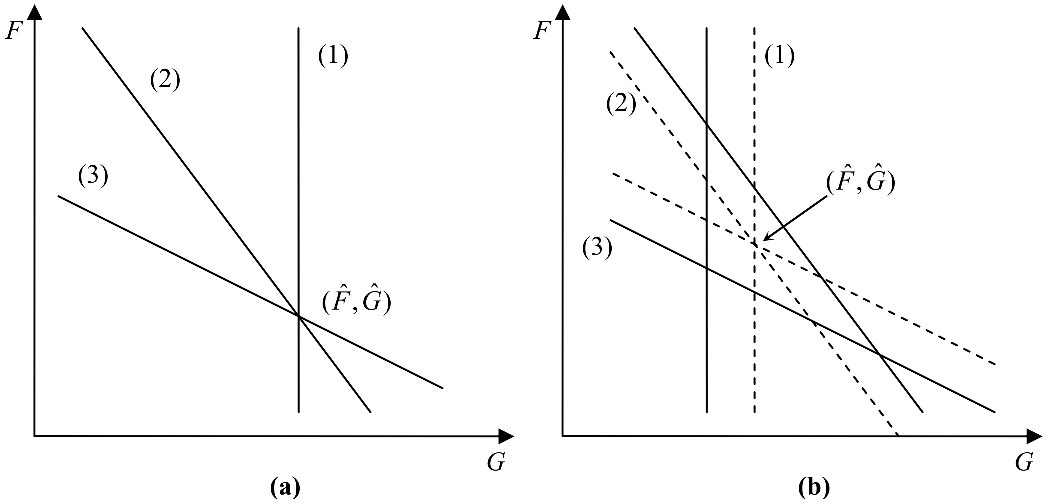


Figure 3. 5 Minimum-maximal error estimation

(a) for accurate frequency estimates; (b) for inaccurate frequency estimates

Figure 3.5 illustrates this idea by using three partials, indexed 1 (fundamental), 2 and 3. The solid lines are zero-error lines for each partial. Points (F, G) on these lines accurately predict the corresponding frequency estimates under the model (3.11a). Equal-error lines for each partial (dashed lines) are parallel to the zero-error line of that partial. The farther a point departs from this solid line, the more discrepancy there is between the model parameters and the frequency estimate. If all frequency estimates perfectly follow the model, all these solid lines intersect at a single point $(\hat{F},\hat{G})$ (Figure 3.5 (a)). This point has a maximal error of zero, which is apparently the minimal maximum. However, if the estimates carry errors, or the true frequencies do not accurately follow the model (3.11a), the zero-error lines usually do not meet at the same point (Figure 3.5 (b)). By allowing an error for each partial, we are able to move these zero-error lines to parallel equal-error lines, so that at some point they may meet together. The smallest allowed error that enables the three equal-error lines to meet together leads to the minimal maximum estimation. Details on the minimal-maximum search for the stiff string model are given in Appendix D.2.

3.2.7 Harmonic partial grouping

Now we address the task of grouping short-time sinusoid atoms into harmonic particles. The frequency content of a harmonic particle is described by R, while the power content is described by the partial amplitudes a^1, a^2, …, a^M. To find a harmonic particle, we choose from a group of pre-detected sinusoid atoms a subset with frequency and amplitude estimates $(\hat{f}^m, \hat{a}^m)_{m=m_1, m_2, \cdots m_M}$, then use these frequency estimates to calculate R. The subset is chosen according to some criteria concerning harmonicity and power. In this section we use the stiff string model for inharmonicity.

3.2.7.1 Grouping with a given frequency range

We start with the simplest case, that the analyzer is given a small range in which lies the fundamental frequency, or any other partial frequency, or a combination of arbitrary partial frequency ranges. The given range is small enough to cover no more than a few spectral peaks. Using these ranges we can initialize a relatively small R with the method in Appendix D. Using this initial R we are able to bracket an interval in which to search for the 1st partial, 2nd partial, etc. Individual partial searching is done by looking up the pre-detected sinusoid atoms. When a partial is found, we can use its frequency estimate to update R. However, there are two complications: that a partial may not appear as a sinusoid atom, and that there may be multiple sinusoid atoms competing for one partial.

A. Competing atoms

Let the frequency interval for the mth partial be (f_-^m, f_+^m). If there is more than one spectral peak lying in this range, then each of them is treated as a *candidate* for the mth partial. Starting from each candidate we may go on looking for further partials, where we may encounter more competing atoms. To reach a decision from multiple candidates, we need a criterion to compare them. Let p_1 and p_2 be two harmonic particle candidates estimated for the same event. Intuitively p_1 is better than p_2 if it captures more correct partials and less incorrect ones than p_2. However, this criterion is not practical for the analyzer as it requires the ground truth. We base our criterion on two assumptions:

(1) Most spurious atoms are weak;

(2) correctly captured partials tend to have less frequency departure from the model frequency.

Assumption (1) favours strong partials. So if atom p₁ has higher power than atom p₂, it is given a higher score on the strength side. The power itself can be the total amplitude, calculated by summing up the partial amplitudes, or a total partial loudness, calculated by summing up the logarithms of partial amplitudes, or some other perceptual measure. In the least-square-error sense, we use the total square amplitude. Whatever measure we use, we write it in the additive form

$$s_a\left(\{\hat{a}^m\}_{m=1,2,\cdots}\right)=\sum_m s_a^m(\hat{a}^m)\,,\ s_a^m(\hat{a}^m)\geq 0\,,\ \frac{ds_a^m(x)}{dx}>0 \tag{3. 27}$$

Assumption (2) favours partials with more predictable frequencies. As said in 3.2.6, relatively large values are selected for Δ^m to comply with unpredictable errors. This is, however, a main reason why we have competing sinusoid atoms. To make up for this, we introduce the harmonicity criterion based on the departure of frequency estimates from the model. The departure of the m^{th} partial frequency estimate $\hat{f}^m$ from model R is

$$d^m(\hat{f}^m,\mathrm{R})=\begin{cases} f_-^m(\mathrm{R})-\hat{f}^m, & \hat{f}^m<f_-^m(\mathrm{R}) \\ 0, & f_-^m(\mathrm{R})\leq\hat{f}^m\leq f_+^m(\mathrm{R}) \\ \hat{f}^m-f_+^m(\mathrm{R}), & \hat{f}^m>f_+^m(\mathrm{R}) \end{cases} \tag{3. 28a}$$

where $f_-^m(\mathrm{R})$ and $f_+^m(\mathrm{R})$ are defined by (3.21). There can be variations to (3.28a), such as recalculating R with tighter error bounds, moving the boundaries of the piece-wise definition, etc. We also need a post-processing of $d(\hat{f}^m,\mathrm{R})$ before combining them into a total measure, so that it is consistent with the strength criterion (3.27) in some sense. That is

$$s_f(\{\hat{f}^m\}_{m=1,2,\cdots},\mathrm{R})=\sum_m s_a^m(\hat{a}^m)\cdot s_f^m\left(d^m(\hat{f}^m,\mathrm{R})\right) \tag{3. 28b}$$

Let Δ^m be the maximal allowable frequency departure, i.e. any frequency departure beyond it is unacceptable. Accordingly we do a 100% penalty to $d^m(\hat{f}^m, \mathrm{R}) \geq \Delta^m$ by cancelling all its contribution to the strength measure, and no penalty to $d^m(\hat{f}^m, \mathrm{R})=0$, i.e.

$$s_f^m(d) = \begin{cases} 0, \quad d = 0 \\ -d/\Delta^m, \quad 0 < d < \Delta^m \\ -1 \quad or \quad -d/\Delta^m, \quad d \geq \Delta^m \end{cases} \tag{3. 28c}$$

Between $d^m(\hat{f}^m, \mathrm{R})=0$ and $d^m(\hat{f}^m, \mathrm{R}) \geq \Delta^m$ we do a partial penalty, which may also take other forms than the linear one in (3.28c), as long as it stays monotonic. The final scoring function that evaluates a harmonic particle is

$$s\left(\{\hat{a}^m, \hat{f}^m\}_{m=1,2,\cdots}, \mathrm{R}\right) = s_a\left(\{\hat{a}^m\}_{m=1,2,\cdots}\right) + s_f(\{\hat{f}^m\}_{m=1,2,\cdots}, \mathrm{R}) = \sum_m s^m(\hat{a}^m, \hat{f}^m, \mathrm{R}) \tag{3. 29a}$$

where

$$s^m(\hat{a}^m, \hat{f}^m, \mathrm{R}) = s_a^m(\hat{a}^m)\left(1 + s_f^m(d^m(\hat{f}^m, \mathrm{R}))\right) \tag{3. 29b}$$

is interpreted as the contribution of the m^{th} partial to the strength-harmonicity score s.

In the absence of competing atoms, the harmonic grouping proceeds as *linear search*, i.e. the partials are found one after another. The presence of competing atoms expands the linear search to *tree search*. A new branch emerges whenever there are competing atoms, producing multiple candidate atoms for that partial, each generates a new incomplete harmonic particle candidate. However, at any stage we can combine two incomplete harmonic particles p_1 and p_2, if 1) $s_1 > s_2$ and 2) $\mathrm{R}_1 \supseteq \mathrm{R}_2$. In particular, at any branching point, let $\hat{f}_1$ and $\hat{f}_2$ be two peaks that compete for the m^{th} partial, $s^m(\hat{a}_1, \hat{f}_1, \mathrm{R}) > s^m(\hat{a}_2, \hat{f}_2, \mathrm{R})$, then we can immediately remove candidate 2 if a) $\hat{f}_1 > \hat{f}_2$ and $\hat{f}_1 - \Delta^m \leq f_-^m(\mathrm{R})$, or b) $\hat{f}_1 < \hat{f}_2$ and $\hat{f}_1 + \Delta^m \geq f_+^m(\mathrm{R})$. The searching stops when there are no more partials to be found. The harmonic particle candidate with the highest score is chosen as the grouping result. We summarize the tree search as follows:

```
Let $\mathrm{R}_1$ be initialized from a given frequency range, $\mathrm{s}_1=0$, $N=1$. For $m=1, 2, 3, \ldots$,
    do 1~5;
1. let $n'=0$; for $n=1, \ldots, N$, do 2~4;
    2. calculate the frequency range for the $m^{\mathrm{th}}$ partial using $\mathrm{R}_n$;
    3. for each pre-detected atom within this frequency range, do 4;
        4. $n' \leftarrow n'+1$; use the atom to update $\mathrm{R}_n$ to $\mathrm{R}'_{n'}$; calculate $\mathrm{s}'_{n'}$ using (3.29a);
(at this point the highest $\mathrm{s}'_{n'}$ corresponds to the best result with $m$ partials)
5. $N \leftarrow n'$; $\mathrm{R}_n \leftarrow \mathrm{R}'_n$, $s_n \leftarrow \mathrm{s}'_n$, $n=1, 2, \ldots, N$.
```

Details on removing redundant candidates are ignored here.

B. Unfound partials

The unfound partial problem has been addressed in 3.2.1. A partial being unfound is not a problem by itself, as its absence does not affect R, or affect the searching of other partials. The problem is that we do not know whether a partial appears as a spectral peak or not. Even when a partial does not produce a spectral peak by itself, it is possible for spurious peaks to appear where it is expected. If this were the case and the peak were used to update R, the searching range of further partials would be biased. A safe way to deal with the unfound partial problem is always reserving a place for an unfound partial candidate, even when an atom or more has been located. In practice this is necessary only when the size of R is relatively large and the found atom has a large model frequency departure (in which case the found partial substantially reduce the frequency ranges of future searching). Unfound partials do not contribute to $s^m(\hat{a}^m, \hat{f}^m, \mathrm{R})$.

The number of candidates grows with each branching. Fortunately R becomes tight quickly after a few partials have been located, so the competing atom problem does not grow out of control. To further control the scale, we process the partials in two stages: in the first stage we process only the partials without competing peaks, and leave the rest to the second stage. Since after the first stage R has become relatively small, the chance for peaks to compete in the second stage is minimized. A simple trimming is done to keep the number of candidates below a given limit. Whenever the number exceeds this limit, the candidate with the lowest score is discarded. When the search is over, we select the remaining candidate with the highest score as the grouping result.

We run tests on synthesized harmonic sinusoids with white noise. The *SNR* ranges from -15dB to 45dB. Each test sample is 44100 samples in length. A Hann window is used for calculating the DFT's, with fixed window size 1024 and hop size 512, so that from each sample we have 85 results. The fundamental frequency ranges from 5 bins to 40 bins (1bin=1/1024),. The amplitudes follow a reciprocal law, i.e. $a^m=1/m$. The performance of harmonic grouping is evaluated by the *peak collection rate*, defined as the number of correctly located sinusoid atoms divided by the total number of atoms. An atom is classified as correctly located if its frequency estimate is within 0.5 bin from the true frequency. Results for constant harmonic sinusoids are given in Table 3.1, with *B* ranging from 0 to 0.001. Results for pitch-modulated harmonic sinusoids are given in Table 3.2, with the modulator amplitude fixed at 1 semitone, and the modulator period T_M ranging from 2 frames to 12 frames (counted by frame hops, i.e. 512 samples).

SNR *B*	-15dB	0dB	15dB	30dB	45dB
0	20.13	63.56	99.83	100	100
0.0002	31.07	75.59	99.98	100	100
0.0004	29.55	82.17	99.98	100	100
0.0006	33.27	79.25	99.99	100	100
0.0008	32.50	84.78	99.99	100	100
0.001	31.12	85.98	100	100	100

Table 3. 1 Peak collection rate for constant harmonic sinusoids with stiff-string inharmonicity and white noise (%)

SNR: signal-to-noise ratio; *B*: stiffness coefficient

SNR T$_M$	-15dB	0dB	15dB	30dB	45dB
2	20.59	26.98	35.00	35.91	36.10
4	28.07	52.48	69.14	73.77	74.03
6	30.77	57.02	74.51	79.46	81.01
8	30.17	60.57	79.30	83.91	84.89
10	31.31	60.72	83.32	87.98	88.91
12	30.39	62.06	84.06	89.44	90.39

Table 3. 2 Peak collection rate for frequency-modulated harmonic sinusoids with white noise (%)

SNR: signal-to-noise ratio; T$_M$: modulator period (in frames, 1 frame=512 samples)

3.2.7.2 Predominant harmonic particle

If no knowledge of any partial frequency is available, we can still use the method in 3.2.7.1, with *L* competing peaks for the fundamental, where *L* is the total number of pre-detected sinusoid atoms. Once an atom is assumed to be the fundamental, the rest of the searching is reduced to the task in 3.2.7.1. In the case that the fundamental does not show up as a spectral peak, all the *L* atoms compete for the 2nd partial, etc. In the end we can always find a harmonic particle that is optimal by the strength-harmonicity criterion. We call it the *predominant harmonic particle*, and its pitch the *predominant pitch*. This is a different definition from [KVH00, Goto04]. However, their physical concepts are very similar.

The procedure stated above is hardly practical due to the high computation load and the lack of constraint on the fundamental (as it's likely to lower the fundamental to collect as many spectral peaks). We start searching from the strongest peak, let its frequency be $\hat{f}_1$. Without knowing its partial index, we run a quick test through all possible indices, i.e. let *m*=1, 2, …, and let $\hat{f}_1$ be $\hat{f}^m$. We calculate a score for each *m* as

$$s_a(m) = \sum_l \left| \sum_k X_k W\left(k - \frac{l}{m}\hat{f}_1\right) \right|^2 \tag{3. 30}$$

$\left|\sum_k X_k W\left(k-\frac{l}{m}\hat{f}_1\right)\right|^2$ is roughly interpreted as the energy of an LSE estimate of a sinusoid atom with fixed partial frequencies $l(\hat{f}_1/m)$, up to a constant factor. For m=2, 3, 4, …, we calculate 1-$s_a(m')/s_a(m)$, where $m'|m$, $m'<m$. If this value is below some threshold, we remove m from the list. Every m that is left in the list provides a pitch candidate $\hat{f}_1/m$. We derive a harmonic particle candidate for each m by initializing a particle searching with $\hat{f}^m=\hat{f}_1$, and select the optimal one according to the strength-harmonicity criterion as the *predominant harmonic particle involving* $\hat{f}_1$. Similarly, we may proceed with the second strongest peak $\hat{f}_2$, third strongest peak $\hat{f}_3$, etc., and finally select the predominant harmonic particle of this frame.

3.2.7.3 Finding harmonic particle in the presence of other particles

Although we can compare two concurrent harmonic particles using some criterion to determine which is more “predominant”, it is more useful if we are able to detect both. A technique closely related to the detection of multiple harmonic particles is multipitch estimation [KVH00, Klapuri01]. In these works the multipitch estimation proceeds in an iterative manner: in every iteration a predominant pitch is detected and removed from the signal, so that it no longer has predominance in the residue. In the context of harmonic sinusoid modeling we run a similar iterative process. In each iteration a predominant harmonic particle is found. However, instead of subtracting the found harmonic particle from the signal, we mark its partials as “used”. On one hand, the “used” tag means we can ignore this spectral peak in further harmonic particle searching, so that we achieve a similar effect as removing a predominant pitch. On the other hand, by preserving these peaks rather than simply deleting them, we allow a used peak to be reused, i.e. shared among harmonic particles. These tags are used for the further searching of concurrent harmonic particles.

We use the score defined in (3.29a). A summary score of multiple harmonic particles, indexed on p, is given as

$$s\left(\{\hat{a}_p^m,\hat{f}_p^m,\mathrm{R}_p\}_{m=1,2,\cdots;p=1,2,\ldots}\right)=\sum_p s\left(\{\hat{a}_p^m,\hat{f}_p^m,\mathrm{R}_p\}_{m=1,2,\cdots}\right)=\sum_p\sum_m s^m(\hat{a}_p^m,\hat{f}_p^m,\mathrm{R}_p) \quad (3.\ 31)$$

A complication is that the arrangement of spectral peaks into harmonic particles that maximizes the left side of (3.31) not necessarily maximizes the addends on the right side. In other words, earlier detected harmonic particles may have incorrectly collected spectral peaks of other harmonic particles. This should be taken care of during the iterative searching for multiple harmonic particles.

Let there be P already detected harmonic particles $p_1, p_2, \ldots, p_P$, and now we look for the $(P+1)^{\text{th}}$. To do this we find the strongest unused spectral peak, let it be $\hat{f}_{P+1}$. Again, we run a quick test through all possible partial indices for $\hat{f}_{P+1}$. Instead of using the score (3.30), we use

$$\tilde{s}_a(m) = \sum_l \tilde{s}_a^l(m),\ \tilde{s}_a^l(m) = \begin{cases} 0, & \text{if } \frac{l}{m}\hat{f}_{P+1} \text{ is within the main lobe of a used peak,} \\ s_a^l(m), & \text{if not.} \end{cases} \tag{3. 32}$$

Equation (3.32) ignores the contribution of already-used peaks. For m=2, 3, 4, …, we calculate $1\text{-}s_a(m')/s_a(m)$, where $m'|m$, $m'<m$. If this value is below some threshold, we remove m from the list. Every m that is left in the list provides a pitch candidate $\hat{f}_{P+1}/m$. We derive a harmonic particle candidate for each m by initializing R_{P+1} with $\hat{f}^m = \hat{f}_{P+1}$. Once R_{P+1} has been initialized, the searching range for the μ^{th} partial becomes $(f_-^\mu(\mathrm{R}_{P+1}), f_+^\mu(\mathrm{R}_{P+1}))$. Regarding the spectral peaks within this interval, we discuss three possibilities:

1) there are only used peaks in the interval,
2) there are both used and unused peaks in the interval, and
3) there is no used peak in the interval.

In the first case the μ^{th} partial either shares a peak with a partial of another harmonic particle, or does not appear as a peak. In the case of sharing spectral peak, we need amplitude estimates contributed from each harmonic particle to evaluate the score (3.31). However, individual amplitudes are not available. To evaluate (3.29a), we take an approximation by assigning the total amplitude to one harmonic particle, and zero amplitude to others. In the case of the $(P+1)^{\text{th}}$ harmonic particle, if the μ^{th}

peak is shared and its amplitude $\hat{a}$ has been assigned to $\hat{a}_p^m$, then we either set $\hat{a}_{P+1}^{\mu}=0$ or $\hat{a}_{P+1}^{\mu}=\hat{a}$ and $\hat{a}_p^m=0$. In both ways $s_a^m(\hat{a}_p^m)$ is added only once in (3.31) with its true amplitude, no matter how many harmonic particles share it; and it contributes the same score no matter to which harmonic particle the true amplitude is assigned to. This is reasonable as the strength of a spectral peak is not affected by whether or not it is being shared among harmonic particles. However, the frequency departure score s_f^m depends on the assignment of the amplitude. Accordingly, we choose to assign the full amplitude to a harmonic particle that maximizes (3.31). In other words, the amplitude is assigned to the harmonic particle for which the frequency of the shared peak has minimal departure. Shared peaks are not used to update R_{P+1}.

In the second case we may 1) assume the μ^{th} partial being a weak partial without a spectral peak, 2) let the μ^{th} partial share a peak with another harmonic particle, 3) let the μ^{th} partial take an unused peak by itself, or 4) replace a peak p from another harmonic particle with an unused peak and assign p to the μ^{th} partial. Let the used peak be $\hat{f}_{old}$, which has been assigned to the m^{th} partial of the harmonic particle p_p, and the unused peak be $\hat{f}_{new}$. We calculate the changes of the score (3.31) associated with the four cases:

1) $\Delta s_1=0$;

2) $\Delta s_2=\max\Big(0,\, s_a^{\mu}(\hat{a}_{old})s_f^{\mu}\big(d^{\mu}(\hat{f}_{old},\mathrm{R}_{P+1})\big)-s_a^m(\hat{a}_{old})s_f^m\big(d^m(\hat{f}_{old},\mathrm{R}_p)\big)\Big)$;

3) $\Delta s_3=s^{\mu}(\hat{a}_{new},\hat{f}_{new},\mathrm{R}_{P+1})$;

4) $\Delta s_4=s^m(\hat{a}_{new},\hat{f}_{new},\mathrm{R}_p)-s_a^m(\hat{a}_{old})s_f^m\big(d^m(\hat{f}_{old},\mathrm{R}_p)\big)+s_a^{\mu}(\hat{a}_{old})s_f^{\mu}\big(d^{\mu}(\hat{f}_{old},\mathrm{R}_{P+1})\big)$.

Δs_2 and Δs_3 are non-negative, while Δs_4 can be positive or negative. By introducing case 4 we create a chance for modifying the already detected harmonic particles in the context of another new harmonic particle. Although we did not explicitly test if $\hat{f}_{new}$ is compatible with R_p, it is easy to test that $\Delta s_4 \le \Delta s_2$ whenever it is not, which implies that we can remove case 4 from consideration immediately.

In the last case the searching of the μ^{th} partial can proceed as if there are no concurrent harmonic particles at all.

Assuming at least one atom is located for the new harmonic particle, we summarize the harmonic grouping in the presence of other harmonic particles as follows.

```
Let $\mathrm{R}_1$ be initialized from a given atom, $\mathrm{s}_1=0$, $N=1$. For $m=1, 2, 3, \ldots$, do 1~6;
1. let $n'=0$; for $n=1, \ldots, N$, do 2~3;
    2. calculate the frequency range for the $m^{\text{th}}$ partial using $\mathrm{R}_n$;
    3. for each pre-detected atom $a$ within this frequency range, do 4 if it is used
       by some harmonic particle $p$, 5 if not;
        4. if there is an unused atom $a'$ so that $a'$ maximizes $\Delta s_4$ and $\Delta s_4>\Delta s_2$, do
           4.1; otherwise do 4.2;
            4.1. (atom reassignment) $n'\leftarrow n'+1$; use $a$ to update $\mathrm{R}_n$ to $\mathrm{R}'_{n'}$;
                 $\mathrm{s}'_{n'}\leftarrow s_n+\Delta s_4$; replace $a$ in $p$ with $a'$;
            4.2. (atom sharing) $n'\leftarrow n'+1$; $\mathrm{s}'_{n'}\leftarrow s_n+\Delta s_2$;
        5. $n'\leftarrow n'+1$; use $a$ to update $\mathrm{R}_n$ to $\mathrm{R}'_{n'}$; calculate $\mathrm{s}'_{n'}$ using (3.29a);
(at this point the highest $\mathrm{s}'_{n'}$ corresponds to the best result with $m$ partials)
6. $N\leftarrow n'$; $\mathrm{R}_n\leftarrow \mathrm{R}'_n$, $s_n\leftarrow \mathrm{s}'_n$, $n=1, 2, \ldots, N$.
```

The partials of the $(P+1)^{\text{th}}$ harmonic particle can also be processed in two stages: in the first stage we only process those in the third case (no used peak in the searching interval) without competing peaks, and leave all other partials to the second stage. In the end the candidate with the highest score is selected as the grouping result of p_{P+1}.

We run a test on synthesized signals containing two harmonic sinusoids, one is the target sinusoid and the other is regarded as noise. The fundamental frequency of the target harmonic sinusoid is fixed at 20 bins, while the interval between the fundamentals of the two harmonic sinusoids ranges from -11 semitones to 11 semitones, save the 0 interval. The *SNR*, measured as the power of the target harmonic sinusoid divided by the power of the noise harmonic sinusoid, ranges from -

10dB to 10dB. The harmonic grouping is performed with pre-set frequency range around the target fundamental (18bin, 22bin). The peak collection rate without/with considering a second harmonic sinusoid is given in Tables 3.3 and 3.4, respectively. By referring to a second harmonic sinusoid we are able to collect more peaks than ignoring it, especially when the disturbance from the second harmonic sinusoid if strong.

SNR *B*	-10dB	-5dB	0dB	5dB	10dB
0	60.96	76.27	92.47	98.77	100
0.0002	62.00	76.93	93.74	99.17	100
0.0004	59.01	74.80	91.82	98.97	100
0.0006	61.56	76.40	92.93	98.76	100
0.0008	63.78	76.06	93.96	99.21	100
0.001	63.81	76.06	93.81	99.16	100

Table 3. 3 Peak collection rate of one harmonic sinusoid in the presence of a second harmonic sinusoid and white noise, without referring to the second one (%)

SNR: signal-to-noise ratio; *B*: stiffness coefficient

SNR *B*	-10dB	-5dB	0dB	5dB	10dB
0	67.04	78.62	92.48	98.77	99.99
0.0002	71.54	80.32	93.73	99.17	100
0.0004	68.13	78.33	91.83	98.97	100
0.0006	71.14	80.31	92.92	98.76	99.98
0.0008	72.82	80.54	93.98	99.21	100
0.001	72.53	80.19	93.83	99.19	100

Table 3. 4 Peak collection rate of one harmonic sinusoid in the presence of a second harmonic sinusoid and white noise, referring to the second one (%)

SNR: signal-to-noise ratio; *B*: stiffness coefficient

3.2.8 Section summary

In §3.2 we have proposed an inequality-based modeling of harmonic partial frequencies, including the model, methods for solving the model, methods for estimating frequency-related parameters from the model, and application of the model in the grouping of spectral peaks (sinusoid atoms) into harmonic particles. The harmonic grouping process is based on a strength-harmonicity criterion, and constrained on a signal frame of data. In the next chapter we will discuss the harmonic grouping in the context of other harmonic particles within the same sinusoid.

3.3 Measuring sinusoids from multiple frames

All the parameter estimation methods we have discussed so far, including the LSE method, extract parameters from a single data frame based on a stationary-sinusoid assumption. In the rest of this chapter we discuss the measurement of sinusoidal parameters from sinusoidal tracks that span multiple frames. This section discusses the estimation of parameters from sinusoidal tracks with rough parameter estimates. The next section discusses the selection of frame size using frequency dynamics information. Both methods require that the sinusoid tracks be known, therefore can only be applied as post-tracking estimation. Sinusoidal tracks are generated from sinusoid tracking [MQ86, Serra97] or harmonic sinusoid tracking (see Chapter 4).

The point of estimating sinusoidal parameters using sinusoidal tracks lies in the fact that good parameter estimates are only available when signal dynamics have been considered. As discussed in Chapter 2, the error-tolerating harmonic particle representation allows the use of rough parameter estimates evaluated without considering local dynamics. The tracking stage (Chapter 4) returns harmonic sinusoidal tracks, from which the parameters can be re-estimated in a time-varying context. In the re-estimation stage each partial of a harmonic sinusoid is treated separately. Partial harmonicity is not considered.

Let $\hat{f}_0$, $\hat{f}_1$, …, $\hat{f}_L$ be the instantaneous frequencies of a sinusoid estimated at points 0, h, …, Lh, where h is the hop size between adjacent frames. Let the frame size be N, and the window function w be symmetric with w_0=w_N=0 (so that the *true* frame size is N-1). The re-estimation takes a synthesis approach: we try to find re-estimates of the sinusoidal parameters, from which we can synthesize a sinusoid that, when subjected to the LSE estimator, yields the original LSE estimates [WS06].

Since parameters are only estimated at a small subset of all samples, the synthesis of a sinusoid always requires interpolation. Let $\{\tilde{f}_l, \tilde{a}_l, \tilde{\varphi}_l\}_{l=0,\ldots,L}$ be the re-estimates, and $f(t)$, φ_n and a_n be the interpolated parameter tracks, $\varphi_n = \varphi_0 + 2\pi\int_0^n f(t)dt$. The sinusoid synthesized from these re-estimates is $a_n\cos\varphi_n$. The LSE frequency estimates evaluated from this sinusoid are given by (3.9c):

$$\hat{f}_l = \frac{\sum_{n=0}^{N-2}\eta_{l,n}\int_0^1 f_l(n+t)dt}{\sum_{n=0}^{N-2}\eta_{l,n}},\ \eta_{l,n} = \sum_{k=n+1}^{N-1}\sum_{m=0}^{n} w_{mk}a_{l,k}a_{l,m}\,\mathrm{sinc}\frac{\Delta\varphi_{l,mk}}{\pi}. \tag{3. 33}$$

where $f_l(t)$=$f(lh$-$N/2$+$t)$, $a_{l,k}$=$a_{lh\text{-}N/2+k}$, $\Delta\varphi_{l,mk}$=$\Delta\varphi_{lh\text{-}N/2+m,lh\text{-}N/2+k}$. Each frequency estimate is a weighted average of the instantaneous frequencies of the frame from which it is estimated. The amplitude estimate, however, can be interpreted as a weighted instantaneous amplitude average only when the frequency is constant. The LSE estimation of the amplitude uses the inner product of the analyzed sinusoid with a reference sinusoid at a central frequency. Any departure of the instantaneous frequency from this central frequency will cause the analyzed signal to go out of phase with the reference signal, and therefore produce smaller estimates. This can be corrected by using a variable-frequency reference sinusoid: let the instantaneous frequency track be $f(t)$, we can estimate the amplitude and phase angle using:

$$\hat{a}e^{j\hat{\varphi}} = \frac{\sum_{n=0}^{N-1} w_n^2 x_n e^{-j2\pi\int_{N/2}^{n} f(t)dt}}{\sum_{n=0}^{N-1} w_n^2} \tag{3. 34a}$$

This is the LSE estimate of the frame [0, N-1] given the frequency $f(t)$. When the frequency track using in (3.34a) is accurate, the $\hat{a}$ calculated with (3.34a) is a weighted average of the instantaneous amplitude, with the weights being w_n^2, i.e.

$$\hat{a} = \frac{\sum_{n=0}^{N-1} w_n^2 a_n}{\sum_{n=0}^{N-1} w_n^2} \qquad (3.\ 34b)$$

Since the LSE frequency estimation is a process of averaging, the re-estimation of frequencies from the LSE estimates becomes one of de-averaging. However, as shown by (3.9c), (3.34a) and (3.34b), the de-averaging of the frequencies requires the amplitude track, and the de-averaging of amplitudes requires the frequency track. In [WS06] we proposed to iteratively solve for the re-estimates. This can proceed as follows.

Let F^i stand for a set of frequency estimates, A^i stand for a set of amplitude estimates, P^i={F^i, A^i}. F^0={$\hat{f}_0$, $\hat{f}_1$, …, $\hat{f}_L$}. For i=1, 2, …, do 1~6, until Δ is below some threshold, or i is above a maximal number of iterations:

1) interpolate the frequency estimates $F^{i\text{-}1}$ as $f^{i-1}(t)$;

2) estimate amplitudes $A^{i\text{-}1}$ using the frequency track $f^{i-1}(t)$ with (3.34a);

3) de-average the amplitudes $A^{i\text{-}1}$ as A^i using (3.34b);

4) interpolate the amplitude estimates A^i as a_n^{i-1};

5) de-average the frequencies $F^{i\text{-}1}$ as F^i using (3.9c);

6) calculate the distance Δ between $P^{i\text{-}1}$ and P^i.

In [WS06] the de-averaging was implemented in an average-subtract procedure. That is, we weight-average the parameter estimates $P^{i\text{-}1}$ which are themselves already weighted averages to get Q^i, then compute P^i=2$P^{i\text{-}1}$-Q^i. A refined implementation of the de-averaging stages is discussed in Appendices E.1 and E.2. A de-variation method, as an alternative for de-averaging, is presented in Appendix E.3.

We run a test on synthesized sinusoids with sinusoid-modulated frequencies. The modulator amplitude A_M ranges from 1 bin to 32 bins (1bin=1/1024); the modulating period T_M ranges from 2 frames to 12 frames. Amplitudes, frequencies and phase angles are estimated using the LSE estimator. Parameter estimation performance is evaluated using a synthesis approach. Two constant sinusoids are synthesized from the true and estimated parameter sets respectively, then the error between the two is compared to the former to produce a noise-to-signal ratio (NSR). One synthesis NSR is calculated from each atom. The average NSR on a set of atoms is obtained by averaging the individual atom NSR's. Finally the reciprocal of this average NSR, which we call an *atom SNR*, is used for evaluating parameter estimation. The results with and without re-estimation are listed in Tables 3.5 and 3.6 respectively.

T_M / A_M	2	4	6	8	10	12
1	5.78	15.49	19.80	22.64	24.74	26.44
2	0.83	9.89	14.06	16.78	18.82	20.49
4	-2.09	5.20	9.02	11.35	13.19	14.74
8	-2.78	1.76	4.83	7.16	8.50	9.70
16	-1.84	-0.40	1.16	3.45	5.22	6.16
32	-0.59	-1.11	-0.95	0.45	1.91	3.22

Table 3. 5 Atom SNR (dB) of standard LSE method on frequency-modulated sinusoids

T_M: modulator period, in frames; A_M: modulator amplitude, in bins

T_M / A_M	2	4	6	8	10	12
1	20.50	31.88	42.11	50.76	43.37	44.83
2	13.60	26.31	36.22	44.81	52.29	57.70
4	10.41	21.76	30.59	39.00	46.36	51.93
8	8.91	18.40	25.69	33.45	40.40	46.11
16	7.87	14.01	22.52	27.33	34.60	41.49
32	-0.25	7.59	16.04	21.80	27.66	33.24

Table 3. 6 Atom SNR (dB) of LSE method with multi-frame re-estimation on frequency-modulated sinusoids

T_M: modulator period, in frames; A_M: modulator amplitude, in bins

3.4 Measuring harmonic sinusoids with multiple resolutions

In this section we discuss the evaluation of sinusoidal parameters using Fourier transforms with multiple resolutions. The resolution is determined by the width the window function used for calculating the spectrum. It has two aspects: a time resolution representing the ability to localize events in time, and a frequency resolution representing the ability of the spectrum to resolve events in frequency. The time resolution is reciprocal to the frequency resolution.

When the signal frequency is constant, the accuracy of frequency estimation is mainly determined by the frequency resolution: the higher the frequency resolution, the less noise and disturbance affect the frequency estimate. However, when the frequency varies with time, the time resolution becomes important. Since the frequency estimate is only a weighted average of the true frequency over time, the better the measurement is localized, the smaller estimation error is expected. We show this as follows.

3.4.1 Instantaneous frequency estimation error

To roughly evaluate the frequency estimation error, we take approximations of (3.9c) by ignoring the factor $\operatorname{sinc}(\Delta\varphi_{mn}/\pi)$ and replacing the sums with integrals, fixing the window centre at 0 and window width at 2*τ*, so the window is supported on (-*τ*, *τ*):

$$\hat{f} \cong \frac{\int_{-\tau}^{\tau}\int_{-\tau}^{\tau} w^2\left(\frac{m}{\tau}\right)w^2\left(\frac{n}{\tau}\right)a(n)a(m)(n-m)\int_m^n f(t)dtdmdn}{\int_{-\tau}^{\tau}\int_{-\tau}^{\tau} w^2\left(\frac{m}{\tau}\right)w^2\left(\frac{n}{\tau}\right)a(n)a(m)(n-m)^2 dmdn} \tag{3. 35a}$$

Let Δ(*t*)=*f*(t)-*f*(0), then we have the frequency estimate error when $\hat{f}$ is assigned to the frame centre:

$$\hat{f} - f(0) \cong \frac{\int_{-\tau}^{\tau}\int_{-\tau}^{\tau} w^2\left(\frac{m}{\tau}\right)w^2\left(\frac{n}{\tau}\right)a(n)a(m)(n-m)\int_m^n \Delta(t)dtdmdn}{\int_{-\tau}^{\tau}\int_{-\tau}^{\tau} w^2\left(\frac{m}{\tau}\right)w^2\left(\frac{n}{\tau}\right)a(n)a(m)(n-m)^2 dmdn} \tag{3. 35b}$$

If we expand Δ(*t*) at 0:

$$\Delta(t) = \sum_{k>0} \frac{f^{(k)}(0)}{k!} t^k \tag{3. 35c}$$

then

$$\int_m^n \Delta(t)dt = \sum_{k>0} \frac{f^{(k)}(0)}{(k+1)!}\left(n^{k+1} - m^{k+1}\right) \tag{3. 35d}$$

so

$$\hat{f} - f(0) \cong \frac{\int_{-\tau}^{\tau}\int_{-\tau}^{\tau} w^2\left(\frac{m}{\tau}\right) w^2\left(\frac{n}{\tau}\right) a(n)a(m)(n-m) \sum_{k>0} \frac{f^{(k)}(0)}{(k+1)!}\left(n^{k+1} - m^{k+1}\right) dmdn}{\int_{-\tau}^{\tau}\int_{-\tau}^{\tau} w^2\left(\frac{m}{\tau}\right) w^2\left(\frac{n}{\tau}\right) a(n)a(m)(n-m)^2 dmdn} \tag{3. 35e}$$

Now let us change the window size to $\alpha\tau$, let the estimate be $\hat{f}_{(\alpha)}$, then

$$\hat{f}_{(\alpha)} - f(0) \cong \frac{\int_{-\tau}^{\tau}\int_{-\tau}^{\tau} w^2\left(\frac{m}{\tau}\right) w^2\left(\frac{n}{\tau}\right) a(n\alpha)a(m\alpha)(n-m) \sum_{k>0} \frac{f^{(k)}(0)}{(k+1)!}\left(n^{k+1} - m^{k+1}\right)\alpha^k dmdn}{\int_{-\tau}^{\tau}\int_{-\tau}^{\tau} w^2\left(\frac{m}{\tau}\right) w^2\left(\frac{n}{\tau}\right) a(n\alpha)a(m\alpha)(n-m)^2 dmdn}$$

(3. 35f)

The right-hand side of (3.35f) is a weighted average of $\sum_{k>0} \frac{f^{(k)}(0)}{(k+1)!}\left(n^{k+1} - m^{k+1}\right)\alpha^k$. Compared to (3.35e), we see that the k^{th} term is amplified by α^k, $k\geq 1$. However, for the linear term $k=1$, since it is odd-symmetric, when multiplied with $a(n\alpha)a(m\alpha)$ only the odd part of the latter contributes to the integral. This introduces a factor of α. Therefore roughly speaking, the frequency estimate error due to signal dynamics grows like α^2 or faster.

3.4.2 Choosing resolution for parameter estimation

Based on the above observation we develop a simple multi-resolution method for post-tracking parameter estimation. After partial tracking, we can estimate local parameter dynamics using parameter estimates from adjacent frames. For example, the frequency dynamics at the l^{th} frame can be estimated from the estimates $\hat{f}_{l-1}$, $\hat{f}_l$ and $\hat{f}_{l+1}$ up to order 2. Let it be

$$f(t) = f(0) + f_1 t + f_2 t^2 \qquad (3.\ 36a)$$

The amplitude dynamics, on the other hand, cannot be reliably extracted from the estimates, since the latter are too sensitive to frequency errors. Instead, we preset a maximal reasonable 1st-order amplitude dynamics coefficient. Let it be

$$a(t) = a(0)(1 + a_1 t) \qquad (3.\ 36b)$$

f_1 evaluates the main odd-symmetric part of f; f_2 evaluates the main even-symmetric part of f. We consider these two parts separately. f_k (k=1 or 2) contributes a term $a(0)^2(1 + a_1 m)(1 + a_1 n)(n - m)\dfrac{f_k}{(k+1)!}\left(n^{k+1} - m^{k+1}\right)$ from (3.35e). For f_1 it is

$$\frac{1}{2} f_1 a(0)^2 (1 + a_1 m)(1 + a_1 n)(n - m)^2 (n + m)\left(n^2 - m^2\right) \sim \frac{1}{2} f_1 a(0)^2 a_1 (n^2 - m^2)^2 \qquad (3.\ 37a)$$

where the right hand side is obtained by discarding the terms that vanish after doing the integral. For f_2 it is

$$\frac{1}{6} f_2 a(0)^2 (1 + a_1 m)(1 + a_1 n)(n - m)^2 (n + m)\left(n^3 - m^3\right) \sim \frac{1}{6} f_2 a(0)^2 (n^4 + m^4) \qquad (3.\ 37b)$$

where in deriving the right-hand side we have also discarded the small term involving a_1^2. The signal dynamics can therefore be measured using

$$D = f_1 a_1 + \kappa f_2, \qquad (3.\ 38a)$$

where κ is a constant given by

$$\kappa = \frac{\int_{-1}^{1}\int_{-1}^{1} w^2(m) w^2(n)\left(n^4 + m^4\right) dm dn}{3\int_{-1}^{1}\int_{-1}^{1} w^2(m) w^2(n)\left(n^2 - m^2\right)^2 dm dn} \qquad (3.\ 38b)$$

Combine (3.38a) and (3.35e) we get

$$\hat{f} - f(0) \cong \frac{0.5 D\tau^2 \int_{-1}^{1}\int_{-1}^{1} w^2(n) w^2(m)(n^2 - m^2)^2 dm dn}{\int_{-1}^{1}\int_{-1}^{1} w^2(m) w^2(n)(n^2 + m^2) dm dn - 2a_1^2\tau^2 \int_{-1}^{1}\int_{-1}^{1} w^2(m) w^2(n) n^2 m^2 dm dn} \qquad (3.\ 39)$$

If we ignore the term involving a_1^2, then the error is roughly proportional to $D\tau^2$. This is consistent to our previous discussions in (3.4.1).

Using the above criterion, we examine the sinusoid tracks to see if any of the previous estimates is calculated from a frame with large dynamics (i.e. $D\tau^2>Th$, where Th is some threshold). If this is the case, the parameters are re-estimated using a finer time resolution, i.e. we choose spectra calculated with the window size $2^{-k}\cdot 2\tau$ so that $D\cdot 2^{-2k}\tau^2<Th$, or $2^{-k}\cdot 2\tau$ is the smallest window size we use. Additional measurement points are inserted to make up the gap left from shortening of windows. The routine that determines the measurement points and the window size associated with each measurement point is given as follows.

```
Let there be $L+1$ points in the list at the beginning, located at $n_0$, …, $n_L$, with
    frequency estimates $\hat{f}_0$, …, $\hat{f}_L$, and window sizes $\tau_0=\tau_1=\ldots=\tau_L=N$, the number
    of resolutions (2-based) $K$;
0. let $k=1$, $\tau=N$;
1. for $l=0, 1, \ldots, L-1$, do 2~3;
    2. if $D(n_l)\cdot\tau^2>Th$, do 3;
        3. insert two new measurement points at $0.5\cdot(n_l+n_{l+1})$ and $0.5\cdot(n_l+n_{l-1})$,
           respectively, associate the window sizes $\tau/2$ with these two points, as well
           as point $n_l$;
4. if no point has been inserted in the loop, then terminate the process, as all the
    measurement points are already found;
5. get frequency estimates at all measurement points with window size $\tau/2$ ;
6. let $L$, $n_l$, $\hat{f}_l$ and $\tau_l$ be redefined for the new point sequence with the inserted
    points;
7. $k\leftarrow k+1$, $\tau\leftarrow\tau/2$, go to step 1 if $k<K$.
```

For harmonic sinusoids the frequency dynamics is proportional to the frequency itself. This results in the use of a high time-resolution for high frequencies, in a manner similar to the wavelets.

We run a test on the same data set as in §3.3, with the window width now ranging between 32 and 1024. The threshold *Th* is set at 1/10240. The results are given in Table 3.7. Some modulation effect with respect to the modulator period is observed. The multi-resolution re-estimation method consistently outperforms the plain LSE method in Table 3.5, and outperforms the single-resolution re-estimation method in Table 3.6 when the frequency dynamics are very high.

T_M A_M	2	4	6	8	10	12
1	44.59	18.06	41.59	26.50	36.82	30.96
2	31.62	13.70	38.15	20.68	32.26	25.18
4	36.00	11.86	42.14	15.44	36.86	19.90
8	34.68	13.69	37.88	13.56	32.23	15.46
16	28.03	14.15	39.94	14.89	36.36	12.62
32	29.57	12.72	33.97	16.00	32.19	16.84

Table 3. 7 Atom SNR of LSE method with multi-resolution re-estimation on frequency-modulated sinusoids

T_M: modulator period, in frames; A_M: modulator amplitude, in bins

It is easy to combine the two re-estimation method by applying the single-resolution method starting from the multi-resolution re-estimates. The result for the same test set is given in Table 3.8. By combining the two methods we achieve good results for both high and low signal dynamics.

T_M / A_M	2	4	6	8	10	12
1	36.30	39.57	44.20	48.45	56.01	51.80
2	33.11	34.30	47.37	44.53	51.29	55.93
4	30.83	28.68	42.07	47.16	52.99	48.91
8	34.68	22.25	37.31	41.78	46.67	47.39
16	28.03	19.27	34.17	41.14	46.05	46.11
32	29.57	16.80	30.90	31.79	41.53	44.56

Table 3. 8 Atom SNR of LSE method with multi-resolution and multi-frame re-estimation on frequency-modulated sinusoids

T_M: modulator period, in frames; A_M: modulator amplitude, in bins

3.5 Summary

In this chapter we have discussed the techniques for local measurements of harmonic sinusoids, including pre-tracking and post-tracking measurements. Pre-tracking measurements are based on a stationary-sinusoid assumption due to the lack of knowledge of signal dynamics. The pre-tracking estimates are improved by post-tracking re-estimation. Two types of re-estimation have been discussed in 3.3 and 3.4, respectively.

Between the pre-tracking estimation and the harmonic tracking stages is the harmonic grouping, which derives harmonic particles from pre-detected peaks. We have proposed an inequality-based representation for harmonic partial frequencies which tolerates frequency errors and inharmonicity. In this representation the frequency contents of a harmonic particle is described as a region in a fundamental-inharmonicity space. This representation will also be used as the starting point in the harmonic tracking stage, which is to be discussed in the next chapter.

Chapter 4

Harmonic sinusoid tracking

This chapter is devoted to the techniques for tracking sinusoidal partials in a harmonic context. As stated in §2.4, a lot has been done on tracking individual sinusoids to form sinusoid tracks. However, in harmonic sinusoid modeling it is necessary to track partials with constraints on partial harmonicity. This is implemented by tracking harmonic particles into harmonic sinusoids, instead of tracking sinusoid partials individually. Like standard sinusoid modeling, the harmonic tracking is based on frequency and amplitude continuity criteria. The frequency continuity criterion is modified to model pitch continuity to prevent spurious pitch jumps within an event. The amplitude continuity is designed to model the variations of both the total amplitude and the amplitude distribution among partials. The latter models the *short-time timbre*, which we assume to be continuous within an event.

Harmonic particles are obtained by harmonic grouping, which has been discussed in §3.2 for a single frame. Just like tracking individual partials undermines partial harmonicity, the harmonic grouping on individual frames may corrupt frequency continuity. Therefore in the tracking stage the harmonic grouping is performed jointly with the tracking: once the harmonic particle at a certain frame is determined, the harmonic grouping in the next frame is performed with reference to the already-found harmonic particle, so that the frequency continuity is preserved in the harmonic grouping.

This chapter is arranged as follows. 4.1 and 4.2 discuss the frequency and amplitude continuity criteria for tracking harmonic sinusoids. 4.3 and 4.4 discuss forward harmonic sinusoid tracking, where 4.3 focuses on tracking individual harmonic sinusoids, and 4.4 focuses on tracking multiple harmonic sinusoids. 4.5

discusses end-point detection. 4.6 discusses the forward-backward tracking method for improved robustness, followed by a conclusion in 4.7.

4.1 Frequency continuity

In this section we discuss the upgrade of frequency continuity criteria to harmonic frequency continuity criteria. The frequency continuity criteria in [MQ86] include a hard criterion and a soft criterion. The hard criterion imposes a maximal frequency jump between consecutive frames, allowing only frequency jumps within a given range. Within this allowed range, the soft criterion favours smaller frequency jumps to larger ones. In the context of harmonic particles, we also develop a hard criterion and a soft criterion, as follows.

4.1.1 Frequency evolution boundaries

The hard criterion imposes a maximal jump on the fundamental frequency f^1. In the previous chapter we have proposed to represent the frequency content of a harmonic particle with an area R in the f^1-*B* space. Let R$_l$ be the f^1-*B* range at frame *l*. Since a harmonic sinusoid physically represents a single musical event, it is reasonable to assume that *B* remains constant within the same track. Therefore when initializing R$_{l+1}$ for harmonic grouping at the (l+1)th frame, we extend R along the f^1 axis by Δ^0 on both sides, where Δ^0 is the maximal fundamental frequency jump allowed between frame *l* and frame *l*+1. However, this extension does not necessarily preserve linearity of R in the *F*-*G* plane. To show this, we recall that there are three types of sides in R: those parallel to the *G* axis, those passing through the origin (0, 0), and those with negative slopes. It is trivial to show that the linearity of the first and second type is preserved by the extension along the f^1 axis. For the third type, we take a side of R in the *F*-*G* plane in the form (see (D.1))

$$F + kG - g = 0, \; k>0, g>0. \qquad (4.\ 1a)$$

Let (F_+, G_+) be the point we get by extending (*F*, *G*) along the f^1 axis by Δ^0. We have

$$\sqrt{F_+} = \sqrt{F} + \Delta^0, \; G_+ = GF_+ / F \qquad (4.\ 1b)$$

Then

$$F_+ + kG_+ - \frac{gF_+}{\left(\sqrt{F_+} - \Delta^0\right)^2} = 0\,. \qquad (4.\ 1c)$$

It is apparent that (4.1c), which describes part of the boundary of the extended R, is not linear. Therefore the R$_{l+1}$ initialized by directly extending R$_l$ is no longer a polygon. Fortunately, the maximal frequency jump Δ^0 does not have to be an accurate value or strictly constant for the whole range of *B*. Accordingly it is reasonable to initialize R$_{l+1}$ by extending the vertices of R$_l$ then calculating the convex hull of these extended vertices. This is shown in Figure 4.1.

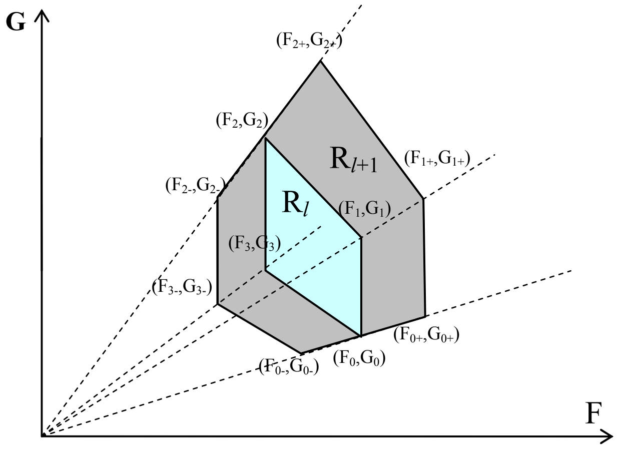


Figure 4. 1 Extending R to allow frequency jump

From (4.1c) we can further calculate

$$\frac{d^2G_+}{dF_+^2} = \frac{3}{2}\frac{g}{k}\frac{1}{\sqrt{F}^3}\left(\frac{1}{\sqrt{F}} - \frac{1}{\sqrt{F_+}}\right) \qquad (4.\ 1d)$$

(4.1d) applies to the third-type sides of the accurately extended R$_l$. According to this equation, such a side is convex when $F_+>F$, i.e. the side is extended by Δ^0, and concave when $F_+<F$, i.e. the side is extended by $-\Delta^0$. These are shown in Figure 4.2 using dashed lines for the R$_l$ in Figure 4.1. It is apparent that the approximate R$_{l+1}$ obtained by directly connecting the extended vertices contains the accurately extended

R_l. In other words, the approximation allows slightly larger positive frequency jump for a range of B, and slightly larger negative frequency jump for another range of B.

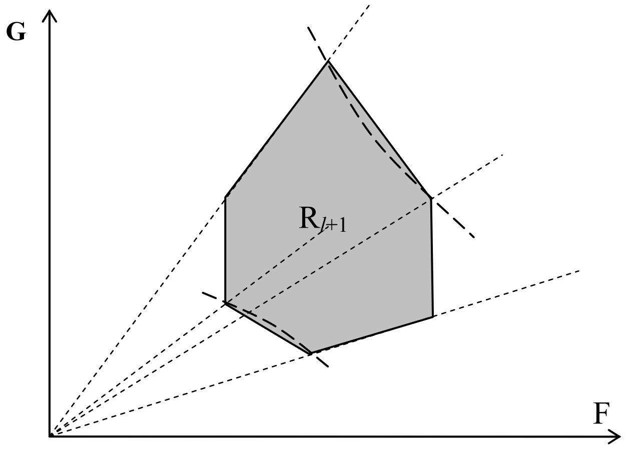


Figure 4. 2 Comparing accurate and approximate extensions of R

The initial R_{l+1} specifies a range to search for partials at frame l+1. The initial size of R_{l+1} depends on the size of R_l and the maximal frequency jump, the latter being typically several bins. The B range may be further reduced with additional partials being located at frame l+1, which, in return affects the previous R_l.

An alternative to the fundamental frequency extension scheme discussed above uses frequency prediction similar to [LMRR03]. That is, instead of confining the fundamental frequency of the (l+1)th frame f_{l+1}^1 within a maximal jump from that of the lth frame, we generate a fundamental frequency prediction for the (l+1)th frame, say f_0, from the previous frames. f_{l+1}^1 is then confined within a maximal jump from f_0. Linear prediction of frequencies encodes the tendency of frequency change. We combine this tendency with the extension of R by applying it to the maximal frequency jumps: to confine f_{l+1}^1 within (f_0-Δ^0,f_0+Δ^0), or equivalently (f_l^1-(Δ^0-f_0+f_l^1), f_l^1+(f_0-f_l^1+Δ^0)), we extend R along the f^1 axis by Δ^0-f_0+f_l^1 on the left side, and by Δ^0+f_0-f_l^1 on the right side.

4.1.2 Comparing fundamental frequency jumps

In standard sinusoid modeling a soft frequency continuity criterion compares frequency jumps to select one from multiple local peaks, all of which fall in the allowed frequency range derived from the hard criterion. Similarly, in harmonic sinusoid tracking we may have competing harmonic particles, whose f^1-B ranges all fall inside the initial R$_{l+1}$. The soft frequency continuity criterion is designed to compare them, so that small fundamental frequency jump is favoured. This is expressed as a continuous function

$$s_f(p_l, p_{l+1}) = 1 - \left|\frac{f_{l+1}^1 - f_l^1}{\Delta_l^0}\right|^p, p\geq 1. \tag{4. 2a}$$

The subscript *f* stands for frequency. Δ_l^0 is the maximal fundamental frequency jump specified between frames *l* and *l*+1. $s_f(p_l, p_{l+1})$>0 when the frequency jump is below Δ^0; $s_f(p_l, p_{l+1})$ <0 when the fundamental frequency jump is beyond Δ^0. *p* is a balancing parameter which controls the amount of penalty done to large frequency jumps. The larger is *p*, the less (4.2a) favours small fundamental frequency jumps against large ones, as long as all jumps are below Δ^0. The frequency predicting version of (4.2a) is

$$s_f(p_l, p_{l+1}) = 1 - \left|\frac{f_{l+1}^1 - f_{l+1}^0}{\Delta_l^0}\right|^p, p\geq 1. \tag{4. 2b}$$

where we have replaced f_l^1 with the prediction f_{l+1}^0. For *p*=2, the scores (4.2a) and (4.2b) are shown in Figures 4.3 (a) and (b) respectively, in solid curves as functions of the fundamental frequency. Spectral peaks are found at the "×" crosses. The scores are positive if the frequency departure from the prediction is smaller than Δ_l, non-positive if not.

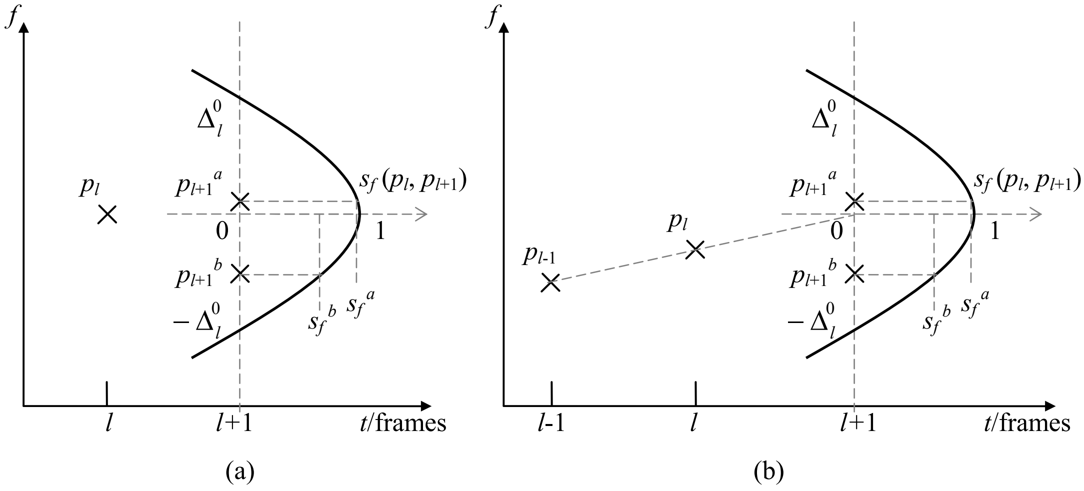


Figure 4. 3 Frequency continuity scores with *p*=2

(a) pure (0th-order) linear prediction; (b) 1st-order linear prediction

Neither of the frequency continuity scores is enough for resolving competing harmonic particles, since there is no guarantee that the next harmonic particle of a harmonic track has to be the one with the smallest fundamental frequency jump in (4.2a), or the one with the least departure from the fundamental frequency predict in (4.2b). Other continuity criteria, which are mostly based on amplitudes, will be considered as no less important in tracking harmonic sinusoids. These are to be discussed in §4.2.

4.2 Amplitude continuity

This section discusses the amplitude continuity criteria used for harmonic sinusoid tracking. Unlike the frequency estimates, which almost always remain within a small interval around the true values, the amplitude estimates may have much larger departures from the true amplitudes. Therefore amplitude continuity is always measured using multiple amplitudes instead of using individual partials, as an amplitude representation summarized from multiple partials tends to be more stable than that derived from individual partials.

Partial amplitudes are used in the plain form or logarithmic form. The log form is calculated by taking the logarithm of the plain amplitudes, floored by a minimal value. That is,

$$a^m \to \begin{cases} \log a^m - \log \varepsilon, & \textit{if } a^m > \varepsilon \\ 0, & \textit{otherwise} \end{cases} \tag{4. 3}$$

4.2.1 Short-term amplitude continuity

The short-term amplitude continuity criterion compares the partial amplitudes of two adjacent frames to measure local continuity. We study two aspects of local amplitude continuity, i.e. 1) the power continuity and 2) the amplitude-ratio continuity. The power is measured by summing up squares of individual partial amplitudes:

$$P = \sum_m \left(a^m\right)^2 \tag{4. 4a}$$

We define the power continuity score as the ratio between the geometric and arithmetic means of the two powers:

$$\left(s_a\right)_1(p_l, p_{l+1}) = \frac{\sqrt{P_l \cdot P_{l+1}}}{\frac{P_l + P_{l+1}}{2}} = \frac{2\sqrt{\sum_m \left(\hat{a}_l^m\right)^2}\sqrt{\sum_m \left(\hat{a}_{l+1}^m\right)^2}}{\sum_m \left(\hat{a}_l^m\right)^2 + \sum_m \left(\hat{a}_{l+1}^m\right)^2} \tag{4. 4b}$$

The subscript "*a*" stands for amplitude. In statistics (4.4b) is used for measuring divergence: the larger the value, the higher the sample similarity. $(s_a)_1$ is amplification invariant, and satisfies $0 \le \left(s_a\right)_1(p_l, p_{l+1}) \le 1$. $(s_a)_1=0$ if the power is zero at either frame; $(s_a)_1=1$ if the power does not change from frame l to frame $l+1$. The amplitude ratio measures the distribution of the total energy among individual partials, which is highly relevant to the short-term timbre. It is measured by the vector of individual partial amplitudes divided by the square root of the power.

$$\mathbf{r} = [r^1\ r^2\ r^3 \cdots]^T, \quad r^m = \frac{a^m}{\sqrt{\sum_m \left(a^m\right)^2}} \tag{4. 5a}$$

We define the amplitude ratio continuity score using their correlation coefficient:

$$\left(s_a\right)_2(p_l, p_{l+1}) = \mathbf{r}_l^T \mathbf{r}_{l+1} = \frac{\sum_m \hat{a}_l^m \hat{a}_{l+1}^m}{\sqrt{\sum_m \left(\hat{a}_l^m\right)^2}\sqrt{\sum_m \left(\hat{a}_{l+1}^m\right)^2}} \tag{4. 5b}$$

This is also known as the *cosine similarity*, as it equals the cosine of the angle between the two amplitude vectors. $(s_a)_2$ satisfies $0 \le (s_a)_2(p_l, p_{l+1}) \le 1$. $(s_a)_2$=0 if the two amplitude vectors are orthogonal; $(s_a)_2$=1 if the two amplitude ratios are identical. A quick combination of (4.4) and (4.5) is

$$s_a(p_l, p_{l+1}) = (s_a)_1 (s_a)_2 = \frac{2\sum_m \hat{a}_l^m \hat{a}_{l+1}^m}{\sum_m \left(\hat{a}_l^m\right)^2 + \sum_m \left(\hat{a}_{l+1}^m\right)^2} \qquad (4.\ 6a)$$

From (4.6a) we can derive the contribution of the m^{th} partial as

$$s_a^m(p_l, \hat{a}_{l+1}^m) = \frac{2\hat{a}_l^m \hat{a}_{l+1}^m}{\sum_m \left(\hat{a}_l^m\right)^2 + \sum_m \left(\hat{a}_{l+1}^m\right)^2} \qquad (4.\ 6b)$$

Again, we have $0 \le s_a(l, l+1) \le 1$; $s_a(p_l, p_{l+1})$=0 if the two amplitude vectors are orthogonal; $s_a(p_l, p_{l+1})$=1 if the two are identical.

Short-term amplitude continuity is used in forward-backward tracking (see §4.6) which requires local scores, as well as in forward tracking (see §4.3) when there are not enough harmonic particles to use the long-term continuity below.

4.2.2 Long-term amplitude continuity

The long-term amplitude continuity criterion compares a harmonic particle at frame *l*+1 with an incomplete harmonic sinusoid suspended at frame *l*, to measure the closeness of the harmonic particle to the harmonic sinusoid. There are two major reasons for using long-term amplitude continuity: the error propagation, and large instantaneous timbre dynamics during pitch variations.

The error propagation is a major risk of using short-term amplitude continuity. Suppose we have two incomplete harmonic sinusoids H^1 and H^2 suspended at frame *l*, with p_{l+1}^1 and p_{l+1}^2 being their correct harmonic particle successors at frame *l*. The short-term continuity criterion connects the correct successor p_{l+1}^1 to H^1 based on the assumption that the correct successor p_{l+1}^1 is locally closer to the current harmonic particle p_l^1 than the incorrect one p_{l+1}^2. However, if it so happened that p_{l+1}^1 and p_{l+1}^2

appear to be so close, that H^1 is extended to the incorrect successor p_{l+1}^2, then it is likely that the tracking of H^1 remains on H^2 after frame l+1, until the same kind of "mistake" brings it back to H^1 again. We refer to this event switching as error propagation. Although the local tracking error at frame l+1 is inevitable when p_{l+1}^1 and p_{l+1}^2 are very close, the use of long-term continuity criterion can prevent error propagation by drawing further tracking back to H^1. This is done by involving previous harmonic particles in the continuity criterion, rather than using p_{l+1}^2 alone.

The amplitude ratio continuity criterion is based on the assumption that different parts of the same harmonic sinusoid are likely to have the same timbre. However, the timbre is more complicated than the amplitude ratio, and may involve the time variation pattern of the amplitude ratio as one aspect. The amplitude ratio itself, on the other hand, may have large local dynamics. The physical model behind these local variations is the formant structure. We assume a source-filter model, in which the instantaneous amplitude can be expressed as the product of the source part, which is a function of partial index, and a filter part, which is a function of instantaneous frequency:

$$a^m = A \cdot A^m \cdot H(f^m),\ \sum_m (A^m)^2 = 1 \tag{4. 7a}$$

where A is an overall amplitude, A^m is the source factor and $H(f)$ is the filter factor. The power is calculated as

$$P = A^2 \cdot \sum_m (A^m)^2 H^2(f^m) \tag{4. 7b}$$

and the amplitude ratio for the mth partial is calculated as

$$r^m = \frac{A^m \cdot H(f^m)}{\sqrt{\sum_m (A^m)^2 H^2(f^m)}} \tag{4. 7c}$$

The filter factor $H(f)$ encodes the formant structure. If we ignore it then the power becomes A^2 and the amplitude ratio becomes A^m, which are the true targets of the power and amplitude ratio continuity criteria. The presence of $H(f)$, combined with the frequency variations, introduces unexpected dynamics to the power and amplitude ratio estimates. This is typically observed in frequency modulated harmonic sinusoids as accompanying amplitude modulations.

Since on the same harmonic sinusoid track the instantaneous frequency of any partial is determined by the instantaneous fundamental and the partial index, we have

$$P = A^2 \cdot \widetilde{P}(f^1),\ r^m = A^m \cdot \widetilde{r}^m(f^1) \tag{4. 8}$$

where $\widetilde{P}(f^1)$ and $\widetilde{r}^m(f^1)$ are functions of the fundamental frequency f^1. Applying (4.8) to (4.4b) and (4.5b) we get

$$(s_a)_1(p_l, p_{l+1}) = \frac{2A_l A_{l+1}\sqrt{\widetilde{P}(f_l^1)\cdot\widetilde{P}(f_{l+1}^1)}}{A_l^2\cdot\widetilde{P}(f_l^1) + A_{l+1}^2\cdot\widetilde{P}(f_{l+1}^1)},$$

$$(s_a)_2(p_l, p_{l+1}) = \sum_m (A^m)^2 \cdot \widetilde{r}^m(f_l^1)\cdot\widetilde{r}^m(f_{l+1}^1) \tag{4. 9}$$

Notice that $(s_a)_2$ only depends on the fundamental frequencies of the two frames, and equals 1 when the fundamentals are identical. When the difference between f_l^1 and f_{l+1}^1 becomes large, the angle between the vectors $\mathbf{r}_l$ and $\mathbf{r}_{l+1}$departs from 0, then $(s_a)_2$ drops from the expected value 1. However, if there is another frame k within the track, and f_k^1 is closer to f_{l+1}^1 than f_l^1 is, then the angle between $\mathbf{r}_k$ and $\mathbf{r}_{l+1}$ will, in general, be smaller than the angle between $\mathbf{r}_l$ and $\mathbf{r}_{l+1}$. This is the basis for defining the long-term amplitude ratio continuity in the *k-nearest-neighbour* (*k*-NN) [CD07] sense.

Let H^1 be an incomplete harmonic sinusoid suspended at frame l, and let l_1, l_2, …, l_k be the indices of k frames of H^1 whose fundamental frequencies are closest to f_{l+1}^1. Then the long-term amplitude ratio continuity is measured as

$$(s_a)_3(H_l, p_{l+1}) = \frac{1}{k}\sum_{\kappa=1}^{k}\mathbf{r}_{l_\kappa}^T\mathbf{r}_{l+1} = \frac{1}{k}\sum_{\kappa=1}^{k}\sum_m r_{l_\kappa}^m r_{l+1}^m \tag{4. 10a}$$

where p_{l+1} stands for the harmonic particle at frame l+1.

For the power continuity criterion, since it is not adequate to assume the power being a function of the fundamental frequency only, there is not a similar nearest neighbour approach. We still use (4.4b) for evaluating power continuity. The long-term version of (4.6) is

$$s_a(H_l, p_{l+1}) = (s_a)_1 (s_a)_3 = \frac{2\sqrt{\sum_m \left(\hat{a}_l^m\right)^2} \cdot \sum_m \left(\frac{1}{k}\sum_{\kappa=1}^{k} r_{l_\kappa}^m\right) a_{l+1}^m}{\sum_m \left(\hat{a}_l^m\right)^2 + \sum_m \left(\hat{a}_{l+1}^m\right)^2} \tag{4. 10b}$$

From (4.10b) we also derive the contribution of the m^{th} in p_{l+1} as

$$s_a^m(H_l, \hat{a}_{l+1}^m) = \frac{2\sqrt{\sum_m \left(\hat{a}_l^m\right)^2} \cdot \left(\frac{1}{k}\sum_{\kappa=1}^{k} r_{l_\kappa}^m\right) \hat{a}_{l+1}^m}{\sum_m \left(\hat{a}_l^m\right)^2 + \sum_m \left(\hat{a}_{l+1}^m\right)^2} \tag{4. 10c}$$

Figure 4.4 compares short-term and long-term continuity criteria, where solid arrows indicate high continuity between atoms, and dashed ones indicate low continuity. Two events are shown in the figures, distinguished by “+” and “×”. The “×” event is the tracking target. The two events become close at frame 4, which renders the harmonic particle at this frame less valid for further tracking. The short-term continuity criterion compares the current harmonic particle with harmonic particles of the next frame to select the best match. Whenever a spurious peak appears to be the most continuous to the current atom, the tracking is diverted to another route, as illustrated in (a) for frames 5 and 6. The long-term continuity criterion, on the other hand, compares candidate harmonic particles with multiple recent frames within the track. We focus on frame 5, where the short-term continuity criterion fails. Even if the current harmonic particle at frame 4 has been corrupted and votes for an incorrect match, the use of frames 1, 2, 3 helps to select the correct one at frame 5.

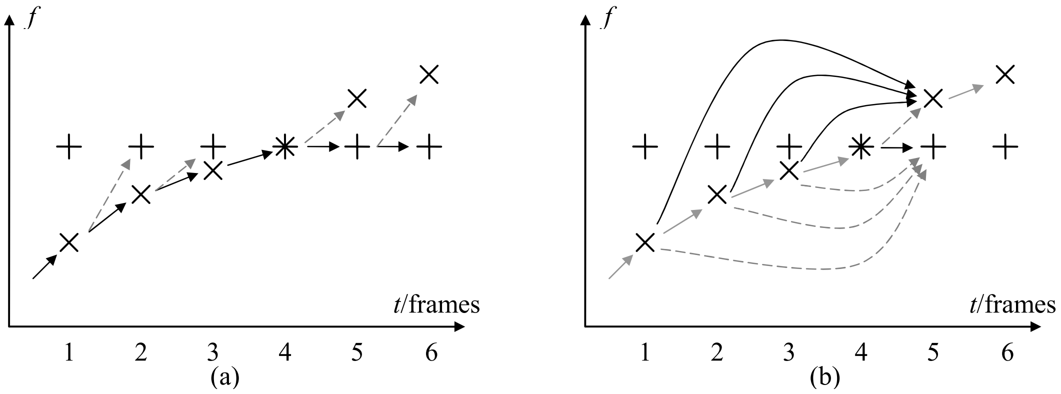


Figure 4. 4 Short-term and long-term continuity

(a) tracking with short-term continuity; (b) tracking with long-term continuity.

Long-term continuity criterion is only used in forward tracking (see §4.3) when there are enough harmonic particles already found, from which the *k* nearest neighbours can be chosen.

4.3 Forward harmonic sinusoid tracking

Starting from a harmonic particle at frame 0, the forward harmonic particle tracking finds a series of incomplete harmonic sinusoids H_l, l=0, 1, 2, …, H_l spanning frames 0, 1, 2, …l, so that $\forall l>0$, H_l contains H_{l-1} and satisfies the frequency and amplitude continuity criteria optimally in some sense. The forward tracking proceeds as follows:

```
Let $\mathrm{H}_0$ include the given harmonic particle at frame 0 only, and $\mathrm{R}_0$ be its $f^1$-$B$
    range. For $l$=1, 2, …, do 1~5,
1) initialize $\mathrm{R}_l$ by extending $\mathrm{R}_{l-1}$ along the $f^1$ axis;
2) find all harmonic particles $p_1, p_2, …, p_n$ using $\mathrm{R}_l$ at frame $l$;
3) if $n$>1, calculate the continuity score between each of them and $\mathrm{H}_{l-1}$, then
    choose the most continuous one, let it be $p_1$;
4) if $n$=0, terminate the loop and return;
5) let $\mathrm{H}_l$ be $\mathrm{H}_{l-1}$ plus $p_1$.
```

This procedure is no more than doing harmonic grouping at a sequence of frames with criteria slightly different from 3.2.7.1, and therefore the same algorithm is used with modifications reflecting the change of criteria, including the initialization of R (step 1, discussed in 4.1.1). Steps 2 and 3 are further explained in 4.3.1 and 4.3.2. The forward tracking finishes if no successor can be found within the range specified by the hard frequency continuity criterion. Like in standard sinusoid modeling, we call this the *death* of a harmonic sinusoid. Other conditions for terminating the harmonic tracking are discussed in §4.5.

4.3.1 Harmonic grouping with predecessors

We have mentioned that apart from those at the starting frame, which are detected without considering the continuity in time, all harmonic particles are grouped with references to other parts of the track. In the context of forward harmonic particle tracking at frame l>0, we refer to previous harmonic particles, or predecessors, i.e. those in $\mathrm{H}_{l\text{-}1}$. In Chapter 3 the harmonic grouping uses a strength-harmonicity criterion for resolving competing peaks, where the strength is based on partial amplitudes and the harmonicity is based on partial frequencies. In the context of forward harmonic tracking, we replace the strength criterion with the amplitude continuity criterion. The strength criterion has been given as

$$s_a\left(\{\hat{a}^m\}_{m=1,2,\cdots}\right)=\sum_m s_a(\hat{a}^m) \tag{3. 27}$$

To apply the amplitude continuity criterion, we replace $s_a\left(\{\hat{a}^m\}_{m=1,2,\cdots}\right)$ in (3.27) with $s_a(p_{l\text{-}1}, p_l)$ in (4.6a) or $s_a(H_{l\text{-}1}, p_l)$ in (4.10b). Otherwise the harmonic grouping process remains unchanged.

4.3.2 Comparing competing harmonic particles

The harmonic grouping does not consider frequency continuity since it does not concern individual partial frequencies. The frequency continuity score is considered after the grouping stage, together with the amplitude continuity scores, to compare multiple harmonic particles that fall in the initial range R_l. For each of the harmonic particle, we calculate the frequency continuity score s_f using (4.2a) or (4.2b), and the amplitude continuity score s_a using (4.6a) or (4.10b). Both scores fall in the range [0, 1], with 0 for poorest continuity and 1 for perfect continuity. The harmonic sinusoid tracker combines the two by taking an average

$$s = \theta \cdot s_a + (1-\theta) \cdot s_f \text{, } 0<\theta<1. \qquad (4.\ 11)$$

The score (4.11) is used in step 3 of the forward tracking process only.

4.3.3 Backward tracking

The backward harmonic sinusoid tracking is the time-reversed version of the forward tracking. Starting from a harmonic particle at frame 0, the backward tracking finds a series of incomplete harmonic sinusoids H_{-l}, l=0, 1, 2, …, H_{-l} spanning frames $-l$, $-l$+1, …0, so that $\forall l>0$, H_{-l} contains H_{-l+1} and satisfies the frequency and amplitude continuity criteria optimally in some sense. Backward tracking progresses in the same way as forward tracking. The combination of forward and backward harmonic sinusoid tracking from the same harmonic particle finds a complete harmonic sinusoid. This technique is used in [WS07] for selecting a harmonic sinusoid from an audio excerpt for editing purposes.

4.3.4 Tests

We run tests on frequency modulated harmonic sinusoids with white noise. The fundamental frequency ranges from 10 bins to 40 bins, the modulator amplitude fixed at 1 semitone, the modulator period ranges from 2 frames to 12 frames, and the *SNR* ranged from -15dB to 45dB. The partial amplitudes follow a reciprocal law regarding instantaneous frequency, so that there is an amplitude modulation accompanying the frequency modulation. Results in Table 4.1 are based on frequency continuity only. Results in Table 4.2 are based on frequency continuity and energy maximization. Results in Table 4.3 are based on frequency and amplitude continuity. It is apparent that the introduction of amplitude criteria helps to improve the tracking. The continuity criterion outperforms the energy maximization criterion when the noise is high, otherwise the two show similar performance. As a comparison, we have also tested the standard sinusoid tracking method, where the atoms in the first frame are initialized using the harmonic particle detector. Frequency and amplitude continuity criterion is applied for comparing competing atoms. The results are given in Table 4.4. All results in Tables 4.1~4.3 are consistently better than those in Table 4.4.

SNR T_M	-15dB	0dB	15dB	30dB	45dB
2	0.83	3.19	9.24	16.99	19.11
4	0.76	4.96	28.34	45.86	53.37
6	0.98	4.15	38.54	71.46	72.66
8	0.83	5.47	36.44	84.22	86.99
10	1.03	4.82	38.22	93.75	94.25
12	1.27	5.45	32.13	96.99	97.21

Table 4. 1 Peak collection rate of tracking frequency-modulated harmonic sinusoid based on frequency continuity only (%)

SNR: signal-to-noise ratio; T_M: modulator period, in frame

SNR T_M	-15dB	0dB	15dB	30dB	45dB
2	4.98	22.27	29.06	25.98	24.96
4	5.94	31.77	53.28	69.44	77.64
6	4.67	34.21	63.97	82.41	82.63
8	4.89	38.09	73.99	92.32	94.07
10	5.47	38.78	80.84	97.32	98.02
12	5.56	41.85	87.51	98.48	98.89

Table 4. 2 Peak collection rate of tracking frequency-modulated harmonic sinusoid based on frequency continuity and maximal strength criterion (%)

SNR: signal-to-noise ratio; T_M: modulator period, in frames

SNR T_M	-15dB	0dB	15dB	30dB	45dB
2	6.12	24.53	28.88	25.60	25.54
4	7.03	31.21	52.02	69.87	73.55
6	7.70	35.38	61.46	76.66	80.80
8	5.78	37.99	77.16	93.53	94.74
10	9.14	39.28	79.77	97.23	97.90
12	7.98	41.78	88.16	98.56	98.96

Table 4. 3 Peak collection rate of tracking frequency-modulated harmonic sinusoid based on frequency and amplitude continuity (%)

SNR: signal-to-noise ratio; T_M: modulator period, in frames

SNR T_M	-15dB	0dB	15dB	30dB	45dB
2	0.56	1.31	2.63	3.88	6.18
4	0.55	1.79	5.21	8.75	21.61
6	0.64	2.22	5.90	11.79	37.58
8	0.62	1.98	5.91	16.82	51.04
10	0.47	1.84	5.20	17.60	66.17
12	0.60	1.84	5.67	20.30	75.48

Table 4. 4 Peak collection rate of tracking frequency-modulated harmonic sinusoid using standard sinusoid analyzer based on frequency and amplitude continuity (%)

SNR: signal-to-noise ratio; T_M: modulator period, in frames

4.4 Forward tracking of multiple harmonic sinusoids

In polyphonic music it is usual to have concurrent pitched events, each of which can be modeled using a harmonic sinusoid. Tracking a harmonic sinusoid without considering the presence of other harmonic sinusoids tends to increase the chance of event switching error. In this section we discuss the method that jointly tracks multiple harmonic sinusoids in the forward tracking framework.

There are two modes for the forward multiple harmonic sinusoid tracking: with or without newborns. Here we have cited the idea of *birth* used in standard sinusoid modeling to refer to the emergence of sinusoids that do not connect to already existing tracks.

4.4.1 Forward tracking without newborns

Given harmonic particles p_0^1, p_0^2, …, p_0^K at frame 0, the forward tracking *without newborns* finds K series of incomplete harmonic sinusoids H_l^k, l=0, 1, 2, …, k= 1, 2, …, K, H_l^k spanning frames 0, 1, …, l, so that $\forall l$, k>0, H_l^k contains H_{l-1}^k, and the K incomplete harmonic sinusoids suspended at frame l jointly satisfy the amplitude and frequency continuity criteria optimally.

Just like the forward tracking of a single harmonic sinusoid being implemented as harmonic grouping for a sequence of frames with continuity criteria, the forward tracking of multiple harmonic sinusoids can be implemented as multiple harmonic grouping, which has been discussed in 3.2.7.3, for a sequence of frames. The successors of H_{l-1}^k, $1 \le k \le K$, are found one after another, each new one is found referring the already-found ones.

The basic idea of the multi-harmonic-particle detection in 3.2.7.3 is the reassignment of already used peaks to other harmonic particles so that the strength-harmonicity criterion is better satisfied. Besides the assignment of peaks to harmonic particles, in harmonic sinusoid tracking each harmonic particle is also assigned to an incomplete harmonic sinusoid as well. Accordingly, it is also necessary to consider reassignments of a second type, which appears as two harmonic sinusoids competing for harmonic particles. In the implementation we ignore the interdependency between the first type and second type assignments and treat them separately. The second type assignments are processed first, which completes a *pitch tracking*. The first type assignments come after, constrained by the results of the pitch tracking, to locate individual partials. The two stages are explained in 4.4.1.1 and 4.4.1.2 respectively.

4.4.1.1 Assigning fundamental frequencies to harmonic sinusoids

Without loss of generality, let H_l^k , k=1, 2, …, K, be ordered by fundamental frequency, H_l^1 being the lowest. At time l+1, the pitch tracking stage finds for each of the k incomplete harmonic sinusoids a successor harmonic particle p_{l+1}^{k}, without considering shared partials. This can be accomplished using the single harmonic sinusoid tracking method discussed in 4.3, as long as the harmonic sinusoids do not compete for successor harmonic particles. In the case a conflict, i.e. the optimal successors of two harmonic sinusoids sharing the same fundamental, is detected, we try to redirect one of them to its sub-optimal successor (the reassignment), so that the overall result is optimal. This proceeds as follows.

```
1. Find the best successors of $H_l^1$, let them be $p_{l+1,j}^{1}$, $j$=1, 2, …, in the order of decreasing continuity (single harmonic sinusoid tracking, see §4.3);
2. for $k$=2, 3, …, $K$, do 3~4;
    3. find the best successors of $H_l^k$, let them be $p_{l+1,j}^{k}$, $j$=1, 2, …, in the order of decreasing continuity;
    4. if $p_{l+1,1}^{k}$ conflicts with any $p_{l+1,1}^{k_1}$, $k_1<k$, i.e. it has the same fundamental frequency as the latter, do 5~6;
    5. if there is neither a $p_{l+1,2}^{k_1}$ nor a $p_{l+1,2}^{k}$, do nothing; if there is $p_{l+1,2}^{k_1}$ but no $p_{l+1,2}^{k}$, do 5.1; if there is a $p_{l+1,2}^{k}$ but no $p_{l+1,2}^{k_1}$, do 5.2; otherwise do 5.3;
        5.1. delete $p_{l+1,1}^{k_1}$ from the successor list of $H_l^{k_1}$, so that $p_{l+1,2}^{k_1}$ becomes the new $p_{l+1,1}^{k_1}$, etc.;
        5.2. delete $p_{l+1,1}^{k}$ from the successor list of $H_l^k$, so that $p_{l+1,2}^{k}$ becomes the new $p_{l+1,1}^{k}$, etc. (reassignment);
        5.3. compute and compare the continuity scores (see (4.11)) $s(H_l^k, p_{l+1,1}^k) + s(H_l^{k_1}, p_{l+1,2}^{k_1})$ and $s(H_l^k, p_{l+1,2}^k) + s(H_l^{k_1}, p_{l+1,1}^{k_1})$; if the previous is higher, then delete $p_{l+1,1}^{k_1}$ from the successor list of $H_l^{k_1}$; otherwise delete $p_{l+1,1}^{k}$ from the successor list of $H_l^k$ (reassignment);
```

6. if any change during 5.1~5.3 causes a new conflict between already found harmonic particles, repeat step 5 on the conflicting pair;

7. let p_{l+1}^{k} be $p_{l+1,1}^{k}$, k=1, 2, …, K.

The reassignment above is based on a continuity-strength criterion. While continuity is given the highest priority, we also aim to connect to the incomplete harmonic sinusoids to as many harmonic particles as possible. Therefore any two of the K harmonic sinusoids are allowed to share a fundamental frequency only when neither of them have a second possible successor. Otherwise we find for each harmonic sinusoid a successor with a distinct fundamental frequency so that the total continuity is optimized (steps 5.1~5.3).

4.4.1.2 Harmonic grouping in the presence of other harmonic particles and predecessors

The harmonic particles found in step 3 in 4.4.1.1 are found independently without considering concurrent harmonic particles. Due to the lack of attention on conflicting harmonic particles in the previous stage, large sinusoidal components may be left unresolved while their nearby components are associated with multiple harmonic particles. This does not satisfy the strength-harmonicity criterion for harmonic particles, and should be corrected by reassignment of atoms, focusing on the atoms on which the harmonic particles conflict.

For comparing different assignments, the strength-harmonicity criterion in 3.2.7.3 is replaced by a continuity-harmonicity criterion. A new issue in the tracking of multiple harmonic sinusoids is the evaluation of amplitude continuity for harmonic particles that share partials. We derive the contribution of the mth partial $\{\hat{a},\hat{f}\}$ of a harmonic particle p_{l+1} to the continuity-harmonicity score by replacing the strength score $s_a^m(\hat{a}^m)$ in (3.27) with a continuity score $s_a^m(H_l,\hat{a}^m)$ as defined in (4.6b) or (4.10c). The short-term continuity version is

$$s^m(p_l,p_{l+1})=s^m\left(p_l,\{\hat{a},\hat{f}\},\mathrm{R}_{l+1}\right)=s_a^m(p_l,\hat{a})\left(1+s_f^m(d^m(\hat{f},\mathrm{R}_{l+1}))\right) \qquad (4.\ 12a)$$

and the long-term version is

$$s^m\left(H_l, p_{l+1}\right) = s^m\left(H_l, \{\hat{a}, \hat{f}\}, \mathrm{R}_{l+1}\right) = s_a^m(H_l, \hat{a})\left(1 + s_f^m(d^m(\hat{f}, \mathrm{R}_{l+1}))\right) \tag{4. 12b}$$

Now suppose a peak $\{\hat{a}, \hat{f}\}$ is shared by two harmonic particles $p_{l+1}^{k_1}$ and $p_{l+1}^{k_2}$, with partial indices m_1 and m_2, respectively, $p_{l+1}^{k_1}$ being assigned to $H_{l+1}^{k_1}$, $p_{l+1}^{k_2}$ to $H_{l+1}^{k_2}$. We propose to split the amplitude $\hat{a}$ into $\hat{a}_1$ and $\hat{a}_2$, and assign $\hat{a}_1$ to $p_{l+1}^{k_1}$, $\hat{a}_2$ to $p_{l+1}^{k_2}$, for the continuity evaluation. The split of $\hat{a}$ follows a simple energy-preservation rule, i.e. $\hat{a}^2 = \hat{a}_1^2 + \hat{a}_2^2$. The portion assigned to each particle is determined by maximizing the total continuity score. For the short-term continuity it is

$$\begin{aligned}
& s^{m_1,m_2}(p_l^{k_1}, p_l^{k_2}, \{\hat{a}, \hat{f}\}, \mathrm{R}_{l+1}^{k_1}, \mathrm{R}_{l+1}^{k_2}) \\
& = s^{m_1}\left(p_l^{k_1}, \{\hat{a}_1, \hat{f}\}, \mathrm{R}_{l+1}^{k_1}\right) + s^{m_2}\left(p_l^{k_2}, \{\hat{a}_2, \hat{f}\}, \mathrm{R}_{l+1}^{k_2}\right) \\
& = s_a^{m_1}(p_l^{k_1}, \hat{a}_1)\left(1 + s_f^{m_1}(d^{m_1}(\hat{f}, \mathrm{R}_{l+1}^{k_1}))\right) + s_a^{m_2}(p_l^{k_2}, \hat{a}_2)\left(1 + s_f^{m_2}(d^{m_2}(\hat{f}, \mathrm{R}_{l+1}^{k_2}))\right) \\
& = A_1\hat{a}_1 + A_2\hat{a}_2
\end{aligned} \tag{4. 13a}$$

where

$$A_1 = \frac{2\hat{a}_l^{k_1,m}\left(1 + s_f^{m_1}(d^{m_1}(\hat{f}, \mathrm{R}_{l+1}^{k_1}))\right)}{\sum_m \left(\hat{a}_l^{k_1,m}\right)^2 + \sum_m \left(\hat{a}_{l+1}^{k_1,m}\right)^2}, \quad A_2 = \frac{2\hat{a}_l^{k_2,m}\left(1 + s_f^{m_2}(d^{m_2}(\hat{f}, \mathrm{R}_{l+1}^{k_2}))\right)}{\sum_m \left(\hat{a}_l^{k_2,m}\right)^2 + \sum_m \left(\hat{a}_{l+1}^{k_2,m}\right)^2} \tag{4. 13b}$$

Ignoring the dependency of A_1, A_2 on $\hat{a}_1$, $\hat{a}_2$, the right-hand side of (4.13a) maximized when $A_1\hat{a}_2 = A_2\hat{a}_1$, with the maximal value

$$s^{m_1,m_2}(p_l^{k_1}, p_l^{k_2}, \{\hat{a}, \hat{f}\}, \mathrm{R}_{l+1}^{k_1}, \mathrm{R}_{l+1}^{k_2}) = \hat{a}\sqrt{A_1^2 + A_2^2} \tag{4. 13c}$$

For long-term continuity this is

$$\begin{aligned}
& s^{m_1,m_2}(H_l^{k_1}, H_l^{k_2}, \{\hat{a}, \hat{f}\}, \mathrm{R}_{l+1}^{k_1}, \mathrm{R}_{l+1}^{k_2}) \\
& = s^{m_1}\left(H_l^{k_1}, \{\hat{a}_1, \hat{f}\}, \mathrm{R}_{l+1}^{k_1}\right) + s^{m_2}\left(H_l^{k_2}, \{\hat{a}_2, \hat{f}\}, \mathrm{R}_{l+1}^{k_2}\right) \\
& = s_a^{m_1}(H_l^{k_1}, \hat{a}_1)\left(1 + s_f^{m_1}(d^{m_1}(\hat{f}, \mathrm{R}_{l+1}^{k_1}))\right) + s_a^{m_2}(H_l^{k_2}, \hat{a}_2)\left(1 + s_f^{m_2}(d^{m_2}(\hat{f}, \mathrm{R}_{l+1}^{k_2}))\right) \\
& = A_1\hat{a}_1 + A_2\hat{a}_2
\end{aligned} \tag{4. 14a}$$

where

$$A_1 = \frac{2\sqrt{\sum_m \left(\hat{a}_l^{k_1,m}\right)^2} \cdot \left(\frac{1}{k}\sum_{\kappa=1}^{k} r_{l_\kappa}^{k_1,m_1}\right)\left(1+s_f^{m_1}(d^{m_1}(\hat{f},\mathrm{R}_{l+1}^{k_1}))\right)}{\sum_m \left(\hat{a}_l^{k_1,m}\right)^2 + \sum_m \left(\hat{a}_{l+1}^{k_1,m}\right)^2},$$

$$A_2 = \frac{2\sqrt{\sum_m \left(\hat{a}_l^{k_2,m}\right)^2} \cdot \left(\frac{1}{k}\sum_{\kappa=1}^{k} r_{l_\kappa}^{k_2,m_2}\right)\left(1+s_f^{m_2}(d^{m_2}(\hat{f},\mathrm{R}_{l+1}^{k_2}))\right)}{\sum_m \left(\hat{a}_l^{k_2,m}\right)^2 + \sum_m \left(\hat{a}_{l+1}^{k_2,m}\right)^2} \qquad (4.\ 14b)$$

Again the maximum is found as $A_1\hat{a}_2 = A_2\hat{a}_1$, where

$$s^{m_1,m_2}(H_l^{k_1}, H_l^{k_2}, \{\hat{a},\hat{f}\}, \mathrm{R}_{l+1}^{k_1}, \mathrm{R}_{l+1}^{k_2}) = \hat{a}\sqrt{A_1^2 + A_2^2} \qquad (4.\ 14c)$$

The above result can easily be generalized to $K(K \geq 3)$ harmonic particles sharing a peak as

$$s_{1\leq j\leq K}^{(m_j)_j}((H_l^{k_j})_j, \{\hat{a},\hat{f}\}, (\mathrm{R}_{l+1}^{k_j})_j) = \hat{a}\sqrt{A_1^2 + A_2^2 + \cdots + A_K^2}\ . \qquad (4.\ 14d)$$

Since harmonic particles are already found in the previous stage, there is no need to start from harmonic grouping as in 3.2.7.3. We proceed with conflicting partials directly. Whenever we detect multiple harmonic particles conflicting at a certain atom, we try to redirect one of them to another atom, so that the overall continuity-harmonicity score is improved.

Suppose at frame l+1 an atom $\{\hat{a},\hat{f}\}$ is shared by harmonic particles $p_{l+1}^{k_1}$, $p_{l+1}^{k_2}$, …, $p_{l+1}^{k_K}$ as $(f_{l+1}^{k_1})^{m_1}$, $(f_{l+1}^{k_2})^{m_2}$,…, $(f_{l+1}^{k_K})^{m_K}$. We calculate the improvement of the score obtained by redirecting $(f_{l+1}^{k_1})^{m_1}$ as follows.

1. Calculate $s_0 = s_{1\leq j\leq K}^{(m_j)_j}((p_l^{k_j})_j, \{\hat{a},\hat{f}\}, (\mathrm{R}_{l+1}^{k_j})_j)$ using (4.14d), which is the score for all K harmonic particles to share $\{\hat{a},\hat{f}\}$;
2. calculate $s_{\bar{1}} = s_{2\leq j\leq K}^{(m_j)_j}((p_l^{k_j})_j, \{\hat{a},\hat{f}\}, (\mathrm{R}_{l+1}^{k_j})_j)$ using (4.14d), which is the score for all K harmonic particles but $p_{l+1}^{k_1}$ to share $\{\hat{a},\hat{f}\}$;

3. let the atoms in the frequency range $(f_{-}^{m_1}(\mathrm{R}_{l+1}^{k_1})-\Delta^{m_1}, f_{+}^{m_1}(\mathrm{R}_{l+1}^{k_1})+\Delta^{m_1})$ other than $\{\hat{a},\hat{f}\}$ be $\{\hat{a}_k,\hat{f}_k\}$, k=1, 2, …; for $\forall k$, calculate the score s_k, contributed by reassigning $\{\hat{a}_k,\hat{f}_k\}$ to $p_{l+1}^{k_1}$, using step 4;

4. if $\{\hat{a}_k,\hat{f}_k\}$ is unused, do 4.1; otherwise do 4.2;

    4.1. calculate s_k=$s^{m_1}(p_l^{k_1},\{\hat{a}_k,\hat{f}_k\},\mathrm{R}_{l+1}^{k_1})$;

    4.2. let $\{\hat{a}_k,\hat{f}_k\}$ be shared by $p_{l+1}^{k'_2}$, …, $p_{l+1}^{k'_{K'}}$ as their m'_2 th, …, $m'_{K'}$ th partials respectively, calculate s_k= $s_{1\le j\le K'}^{(m'_j)_j}((p_l^{k'_j})_j,\{\hat{a}_k,\hat{f}_k\},(\mathrm{R}_{l+1}^{k'_j})_j)$ - $s_{2\le j\le K',\cdots}^{(m'_j)_j}((p_l^{k'_j})_j,\{\hat{a}_k,\hat{f}_k\},(\mathrm{R}_{l+1}^{k'_j})_j)$, where k'_1=k_1, m'_1=m_1;

5. let k^{max}=$\arg\max_k s_k$; if $s_{k^{\max}}+s_{\bar{1}}-s_0\le 0$, then redirecting $(f_{l+1}^{k_1})^{m_1}$ does not bring any improvement; otherwise by redirecting $(f_{l+1}^{k_1})^{m_1}$ to $\{\hat{a}_{k^{\max}},\hat{f}_{k^{\max}}\}$ we improve the overall score by $s_{k^{\max}}+s_{\bar{1}}-s_0$.

The reassignment of atoms is done by repeating the above procedure for all conflicting atoms until no improvement can be achieved by further reassignment.

We test the algorithm on the same test set as in 3.2.7.3 for grouping multiple harmonic particles. The results are given in Table 4.5. In 3.2.7.3 the results in Table 3.4 were generated using the true but rough frequency tracks. The results here show similar performance as those in Table 3.4, indicating that the partial tracking does not go astray frequently.

SNR *B*	-10dB	-5dB	0dB	5dB	10dB
0	72.92	86.00	94.25	99.89	100
0.0002	80.21	88.08	94.55	99.89	100
0.0004	71.42	85.56	93.77	98.89	100
0.0006	73.71	85.78	92.08	98.89	100
0.0008	76.74	87.06	93.25	99.89	100
0.001	76.48	86.43	91.94	99.89	100

Table 4. 5 Peak collection rate of tracking two harmonic sinusoids (%)

SNR: signal-to-noise ratio; T_M: modulator period, in frames

4.4.2 Forward tracking with "births"

In practice few pitched sounds last forever. Accordingly almost all harmonic sinusoids have a beginning and an end. In standard sinusoid modeling an end point of a sinusoid is marked as *birth* or *death*. Typically a sinusoid is "born" when a spectral peak does not have a predecessor, and "killed" when it does not have a spectral peak as a successor. We handle the birth of a harmonic sinusoid in a similar way: a harmonic sinusoid is born if a harmonic particle is found, which does not have a harmonic sinusoid as a predecessor.

New harmonic particles are detected from unassigned peaks using the strength-harmonicity criterion in 3.2.7.3. Only those new harmonic particles that are strong enough are considered as newborns. A hard thresholding is used to qualify new harmonic particles. For harmonic sinusoids with weak starts, the beginning may not qualify for a newborn, so the events are detected some point after the true onsets. All newborns are tracked backwards to locate their beginning points (see §4.5). The backward tracking proceeds in the same way as forward tracking. A *death* point in backward tracking marks the *birth* of a harmonic sinusoid.

4.5 Endpoint detection for harmonic sinusoid tracking

As mentioned above, there are two endpoints for each harmonic sinusoid: a beginning and an end. We always assume that all partials of a harmonic sinusoid start and end at the same time. Although for events with impulsive stimuli it is usual for high frequency partials to fade much quicker than low-frequency ones, we can still tag the faded partials with zero amplitude, so that we can terminate all partials simultaneously.

The end point of a harmonic sinusoid can be located where no successor harmonic particle can be found in forward tracking, and the starting point can be located where no successor harmonic particle can be found in backward tracking. These conditions are, however, usually too weak to terminate events due to the existence of noisy sinusoid peaks and concurrent harmonic sinusoids, i.e. we can almost always find some spurious harmonic particle as a successor to a pending harmonic sinusoid. Unlike the true harmonic particles, spurious harmonic particles consist of spurious peaks or peaks randomly spotted from harmonic events. From a perceptive point of view, a spurious harmonic particle, which is incorrectly attached to an already-finished harmonic sinusoid, will not sound like a natural extension of the event: it either sounds different, or isn't audible at all. We develop a strength thresholding for the latter, and the continuity thresholding for the former. Harmonic particles that fail these thresholdings are no longer considered eligible successors. This helps to terminate harmonic sinusoids at proper points.

4.5.1 Strength thresholding

In strength thresholding we look at the strength of a new harmonic particle p_{l+1} found as a successor to a pending harmonic sinusoid H_l. Whenever the strength falls below a threshold level, p_{l+1} is regarded as ineligible as a successor to H_l.

The threshold is set regarding the strongest point on the harmonic sinusoid. That is, let the strongest frame of H_l have strength A, then the threshold is set at $A\cdot 10^{-Th/10}$, where Th is a relative decay, in dB. For example, when Th=50, then a harmonic sinusoid is terminated whenever the instantaneous strength falls 50dB below its strongest point.

Strictly speaking the strength threshold does not strictly apply to slow-fading sounds, which may accumulate very large decay over a long time before fading off. A moving maximal strength $A_{l-k,l}$ is defined as the maximal strength of *k*+1 frames between frames *l*-*k* and *l*. By defining the threshold at $\max(\varepsilon, A_{l-k,l}\cdot 10^{-Th/10})$, where *ε* is a minimal perceptible threshold, we can track a slow-fading event straight down to the end where it falls out of audible range.

4.5.2 Continuity thresholding

In continuity thresholding we look at the long-term amplitude continuity of a new harmonic particle p_{l+1} found as a successor to an incomplete harmonic sinusoid H_l.

The amplitude continuity score lies between 0 and 1. It may seem plausible to set a threshold for the amplitude continuity score and terminate the harmonic sinusoid whenever the score falls below a threshold level. However, in the context of polyphonic music, an amplitude estimate may represent more than one sinusoid of very close frequencies, which corrupts the computed amplitude continuity score. A typical example is the outburst of new notes that have signification overlap on the partials of p_{l+1}. Due to the symmetrical nature of the continuity score, gaining such significant energy will appear in the score the same as losing the energy, the latter being a typical indicator of an end point.

To solve this problem we use an old-plus-new model to calculate the amplitude continuity score, in which only a part of the partial amplitude are used so that the score is maximized. That is, we look for $\hat{a}^m$, *m*=1, 2, 3, …, so that $0 \le \hat{a}^m \le \hat{a}_{l+1}^m$, and the score

$$s_a(H_l, p_{l+1}) = \frac{2\sqrt{\sum_m \left(\hat{a}_l^m\right)^2} \cdot \sum_m \left(\frac{1}{k}\sum_{\kappa=1}^{k} r_{l_\kappa}^m\right)\hat{a}^m}{\sum_m \left(\hat{a}_l^m\right)^2 + \sum_m \left(\hat{a}^m\right)^2} \equiv \frac{\sum_m B^m \hat{a}^m}{A + \sum_m \left(\hat{a}^m\right)^2} \qquad (4.\ 15)$$

is maximized, where $A = \sum_m \left(\hat{a}_l^m\right)^2$, $B^m = 2\sqrt{A}\left(\frac{1}{k}\sum_{\kappa=1}^{k} r_{l_\kappa}^m\right) = 2\sqrt{A}r^m$, m=1, 2, ….

Unconstrained maximum of (5.15) is obtained by letting

$$\frac{\partial s_a(H_l, p_{l+1})}{\partial \hat{a}^m} = \frac{\left(A + \sum_n \left(\hat{a}^n\right)^2\right)B^m - \left(\sum_n B^n \hat{a}^n\right)2\hat{a}^m}{\left(A + \sum_n \left(\hat{a}^n\right)^2\right)^2} = 0\text{, } m\text{=1, 2, ….} \qquad (4.\ 16a)$$

The solution is

$$\hat{a}^m = \frac{\sqrt{A}B^m}{\sqrt{\sum_m (B^m)^2}} = \frac{\sqrt{A}r^m}{\sqrt{\sum_m (r^m)^2}}\text{ , } m\text{=1, 2, …,} \qquad (4.\ 16b)$$

and the maximum is $\sqrt{\sum_m (r^m)^2}$, which is a value close to 1. This implies that if

$$\hat{a}_{l+1}^m \geq \sqrt{A}\frac{r^m}{\sqrt{\sum_m (r^m)^2}}\text{, } \forall m, \qquad (4.\ 17)$$

then the harmonic sinusoid will not terminate at frame *l*.

If (4.17) does not hold for any *m*, then the maximum of (4.15) is a constrained one. Since $\partial s_a(H_l, p_{l+1}) / \partial \hat{a}^m > 0$ whenever $\hat{a}^m = 0$, at the maximum of (4.15) we either have $\partial s_a(H_l, p_{l+1}) / \partial \hat{a}^m = 0$, or have $\hat{a}^m = \hat{a}_{l+1}^m$ and $\partial s_a(H_l, p_{l+1}) / \partial \hat{a}^m > 0$. We calculate the derivative (4.16a) at $\hat{a}_{l+1}^m$, *m*=1, 2, …. Let N₊ be the set of indices of all partials that have positive partial derivatives, and N₋ be the set of indices of all partials that have non-positive partial derivatives, i.e.

$$\left.\frac{\partial s_a(H_l, p_{l+1})}{\partial \hat{a}^m}\right|_{\hat{a}^m = \hat{a}_{l+1}^m} > 0\text{ , }\left.\frac{\partial s_a(H_l, p_{l+1})}{\partial \hat{a}^n}\right|_{\hat{a}^n = \hat{a}_{l+1}^n} \leq 0\text{, } \forall m \in \text{N}_+, n \in \text{N}_-. \qquad (4.\ 18)$$

We try to find a maximum using the conditions $\hat{a}^m = \hat{a}_{l+1}^m$ and $\partial s_a(H_l, p_{l+1}) / \partial \hat{a}^n = 0$, $\forall m \in$N₊, $n \in$N₋. The solution is

$$\hat{a}^n = \frac{B^n}{\sum_{n\in N_-}(B^n)^2}\left(\sqrt{B^2+\widetilde{A}\sum_{n\in N_-}(B^n)^2}-B\right),\ n\in N_- \quad (4.19)$$

and the maximum is

$$s_a(H_l, p_{l+1}) = \frac{B+\sqrt{B^2+\widetilde{A}\sum_{n\in N_-}(B^n)^2}}{2\widetilde{A}}, \quad (4.20)$$

where $\widetilde{A} = A + \sum_{m\in N_+}(\hat{a}_{l+1}^m)^2$. We verify if $\hat{a}^n \le \hat{a}_{l+1}^n$, $\forall n\in N_-$ and $\frac{\partial s_a(H_l, p_{l+1})}{\partial \hat{a}^m} > 0$, $\forall$ $m\in N_+$. If yes, the amplitude continuity score is found as (4.20). Otherwise, we define $N_-^1=\{n \mid n\in N_-, \hat{a}^n \le \hat{a}_{l+1}^n\}$, $N_-^0 = N_-\backslash N_-^1$, $N_+^1=\{m \mid m\in N_+, \frac{\partial s_a(H_l, p_{l+1})}{\partial \hat{a}^m} > 0\}$, $N_+^0 = N_+\backslash N_+^1$, then update N_- and N_+ with $N_-\leftarrow N_-^1\cup N_+^0$, $N_+\leftarrow N_+^1\cup N_-^0$, and repeat the process above.

4.6 Forward-backward tracking

One drawback of the forward tracking of harmonic sinusoids is that all tracking decisions are made *locally*. Whenever two concurrent harmonic particles at frame l+1 compete for a successor p_{l+1} of a pending harmonic sinusoid H_l, the decision is made immediately without referencing into the future; neither is there a chance to cancel an incorrect assignment of p_{l+1} when the error becomes obvious some frames later. In forward-backward tracking, multiple harmonic particles are reserved at each frame as *candidates*. A harmonic sinusoid is tracked out of these candidates globally to avoid making too-early decisions.

The forward-backward scheme [ASP91] is well know in dynamic programming methods, such as the dynamic time warping and hidden Markov models. In a forward-backward tracking we look for a sequence (s_1, s_2, …, s_L), $s_1\in S_1$, $s_2\in S_2$, …, $s_L\in S_L$, so that a cost function $c(s_1, s_2, …, s_L)$ is maximized or minimized, where the sets S_1, S_2, …, S_L are known a priori. The cost function is supposed to be *cumulative*, i.e.

$$c(s_1, s_2, …, s_L)=c(s_1, s_2, …, s_l)\oplus c(s_l, s_{l+1}, …, s_L), \quad (4.21a)$$

where the scalar operator $\oplus$ satisfies associativity and preserves monotonicity:

$$c(x_1)\oplus c(x_2)\geq c(y_1)\oplus c(y_2),\text{ if } x_1>y_1 \text{ and } x_2>y_2. \qquad (4.21b)$$

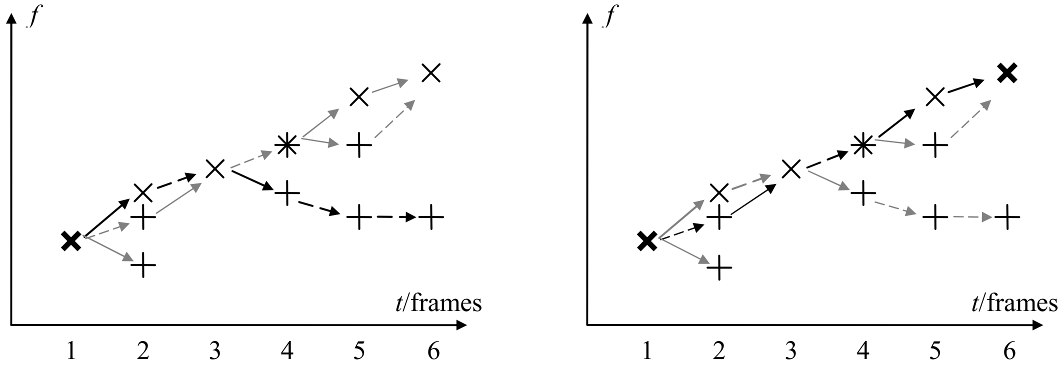
For s_l, $c(s_1, s_2, \ldots, s_l)$ is known as the *backward* cost function and $c(s_l, s_{l+1}, \ldots, s_L)$ the *forward* cost function. According to the definition of $\oplus$, to find the optimal sequence for frames 1, 2, …, L given s_l, one only need to find two optimal subsequences for frames 1, …, l and l, …, L independently. The global optimal sequence if found by trying out all $s_l\in S_l$.

We apply the forward-backward tracking in finding a harmonic sinusoid between two given harmonic particles at the ends. That is, we have $S_1=\{p_1\}$, $S_L=\{p_L\}$, and look for p_2, …, p_{L-1}, so that the tracking criterion is satisfied optimally by the harmonic particle sequence p_1, …, p_L. This task specification comes from the application of harmonic sinusoids for audio editor purposes, in which a user is allowed to specify two points in the time-frequency plane, from which a harmonic sinusoid is tracked out.

Like in the multiple harmonic sinusoid tracking, we separate the pitch tracking and partial tracking in two stages: the pitch tracking is performed first, then the partial tracking is performed with reference to the pitch tracking result. Each of the two stages is a forward-backward process. In pitch tracking the state set S_l contains pitch candidates for frame l, while in partial tracking the state set S_l contains harmonic particle candidates for the chosen pitch at frame l. By ignoring the dependency of pitch tracking on partial tracking results, we get a sub-optimal harmonic sinusoid instead of an optimal one, at the benefit of limiting the size of the state sets S_l, l=1, …, L.

Figure 4.5 compares forward and forward-backward tracking. Target harmonic particles are given as "×", and spurious ones as "+". Strong/weak local continuities between harmonic particles are pictured as solid/dashed arrows. The final route is pictured in dark colour. The target event is corrupted in frame 4, which leads to its weak continuity to frame 3. In forward tracking a connection is made from the current incomplete track to the most continuous harmonic particle, short-term or long-term, in the next frame. Therefore if the most continuous harmonic particle is a spurious one, the tracking fails at this frame. This error may propagate, as shown in (a) after frame 4.

In forward-backward tracking, the continuity is optimized globally, so that the strong continuity between target harmonic particles in frames 4~6 may "make up" for the weak continuity between frames 3 and 4. If the harmonic particles can be externally specified, e.g. for frames 1 and 6, then the error in Figure 4.5(a) can be safely avoided. The emphasis on global optimization may lead to local errors such as at frame 2 in Figure 4.5(b). However, in the long run it is safer than local optimization as long as the cost function is well chosen.



(a)

(b)

Figure 4. 5 Forward and forward-backward tracking

(a) forward searching; (b) forward-backward searching

4.6.1 Cost function

We express the cost function in an cumulative form as

$$c(s_1, s_2, \cdots s_L) = \sum_{l=1}^{L-1} c(s_l, s_{l+1}) \qquad (4.\ 22)$$

Two scoring functions in the form of $c(s_l, s_{l+1})$ are the short-term amplitude continuity score $s_a(p_l, p_{l+1})$ and the fundamental frequency continuity score $s_f(p_l, p_{l+1})$. The two can be combined together as (4.11), which is also in the form of $c(s_l, s_{l+1})$. In our implementation we simply choose $c(s_l, s_{l+1}) = s(p_l, p_{l+1})$. An alternative is taking the logarithm: $c(s_l, s_{l+1}) = \log s(s_l, s_{l+1})$.

4.6.2 Forward-backward pitch tracking

Starting from k_1=1, $p_1{}^1$=p_1, the forward-backward tracking is performed using a standard Viterbi algorithm as follows.

```
1. For $l$=2, 3, …, $L$-1, do 2~5;
    2. for $k$=1, …, $k_{l-1}$, do 3;
        3. initialize $\mathrm{R}_l^k$ from $\mathrm{R}_{l-1}^k$, and find successors $\pi_1$, $\pi_2$, …, of $p_{l-1}^k$ using $\mathrm{R}_l^k$,
           and calculate $s(p_{l-1}^k, \pi_j)$, and set prev($\pi_j$)= $p_{l-1}^k$, $j$=1, 2, … ;
    4. add all harmonic particles found in 3 into a list $(p_l^j)_{j=1,2,\cdots}$;
    5. if any two successors in $(p_l^j)_j$, say $p_l^{j_1}$, an successor of $p_{l-1}^{k_1}$, and $p_l^{j_2}$, an
       successor of $p_{l-1}^{k_2}$, have the same fundamental frequency, then delete $p_l^{j_1}$
       if $s(p_{l-1}^{k_1}, p_l^{j_1}) \le s(p_{l-1}^{k_2}, p_l^{j_2})$, or delete $p_l^{j_2}$ if $s(p_{l-1}^{k_2}, p_l^{j_2}) < s(p_{l-1}^{k_1}, p_l^{j_1})$;
6. initialize $\mathrm{R}_{L-1}$ from $\mathrm{R}_L$, for $k$=1, …, $k_{L-1}$, calculate $s(p_{L-1}^k, p_L)$ if $p_{L-1}^k$ falls in
   the range given by $R_{L-1}$, and choose the largest one, let it be $p_{L-1}$;
7. for $l$=$L$-1, …, 2, let $p_{l-1} = prev(p_l)$.
```

Steps 1~5 proceeds forward, while step 6 proceeds backward. The forward tracking and backward tracking meet at frame L-1. Since the frequency range of the states tends to increase with each forward or backward step, and since there are L-2 forward steps and only 1 backward step, at the meeting frame L-1 the forward range is likely to be much larger than the backward range. As only the candidates within both ranges can contribute to the final result, a large part of $(p_{L-1}^j)_{j=1,2,\cdots}$ is actually useless. To help reducing unnecessary computation we can move the meeting point backward, say to frame l. The forward searching is performed on frames 1 to l, and the backward searching is performed on frames L to l.

4.6.3 Forward-backward partial tracking on given pitch track

Now we consider multiple harmonic particles which have the same fundamental at frame *l*. In the pitch tracking algorithm in 4.6.2 a decision is made immediately to keep at most one of them in step 5. This is a local decision based on short-term continuity between two frames. By making this early decision we keep tight control of the size of S_l during the tracking, at the cost of getting a sub-optimal results of the harmonic particles. To find optimal harmonic particles on the pitch track, we reselect harmonic particles for all the frames in the forward-backward framework so that the cost function (4.22) is optimized. The algorithm is almost the same as the pitch tracking, except that in step 3 we initialize R_l^k from R_{l-1}^k and the fundamental f_l^1, and the condition in step 5 is modified from identical fundamental to identical harmonic particle.

SNR T_M	-15dB	0dB	15dB	30dB	45dB
2	14.65	22.54	26.74	20.33	20.04
4	17.68	32.56	55.24	76.08	83.31
6	18.49	37.04	57.61	85.29	93.13
8	18.45	38.72	72.87	94.67	99.46
10	18.67	40.84	79.92	97.99	99.96
12	18.43	42.77	85.05	99.20	100

Table 4. 6 Peak collection rate of tracking frequency-modulated harmonic sinusoid based on frequency and amplitude continuity using forward-backward tracking (%)

SNR: signal-to-noise ratio; T_M: modulator period, in frames

We run a test on forward-backward tracking using the same test set as in §4.3, where we tested forward tracking only. The results are given in Table 4.6. The forward-backward method has obvious advantage over the forward method when the noise level is high. It is also interesting to compare these results to those in Table 3.2, which were obtained using true but rough fundamental tracks. At high noise levels and fast modulation rates the tracking method shows less peak collection due to tracking errors. However, when the noise level is low and modulation is slow, the tracking results are better than in Table 3.2, which can be credited to the forced continuity in inharmonicity and amplitudes.

4.7 Summary

In this chapter we have discussed the harmonic tracking criteria and algorithms. The tracking is mostly based on frequency and amplitude continuities. The polygonal representation in Chapter 3 is used in harmonic sinusoid tracking to specify the searching range, while several continuity scores are used for comparing the candidates that fall in this range. The forward tracking algorithm can be regarded as an extended version of the tracking method in standard sinusoid modeling, with special consideration on colliding partials. In the end we proposed to use forward-backward method for the specific task of finding a harmonic sinusoid to fill the gap between two given harmonic particles.

Chapter 5

Applications

This chapter proposes several applications of the harmonic sinusoid model. There are two main mechanisms that enable the applications: extraction and parameterization. *Extraction* refers to the retrieval of a pitched event from a mixture of events as an isolated sound. By extraction we are able to perform an operation, which is originally designed for a single pitched event, on an event within a mixture. *Parameterization* refers to the quantization of physical or musical properties of a pitched event into sinusoidal parameters. By studying these parameters it is possible to extract mid- or high-level information that is not directly available from the waveform; by modifying these parameters it is possible to modify the audio object by its mid- or high-level properties.

This chapter is arranged as follows. In 5.1 we discuss the resynthesis of harmonic sinusoids from harmonic sinusoidal parameters, then use it for evaluating harmonic sinusoid modeling. In 5.2 we present the application of harmonic sinusoids in audio editors, which enables some functionality not available in conventional editing tools. In 5.3 we propose several other possibilities for applying harmonic sinusoid models, followed by a conclusion in 5.4.

5.1 Harmonic sinusoid resynthesis

Since the conversion of a harmonic sinusoid model to a sinusoid model is trivial, a harmonic sinusoid can be resynthesized using the sinusoid resynthesizer discussed in §2.5. The general theories of the synthesis have already been discussed in Chapter 2. This section only considers some specific issues in the synthesis following the re-estimation (§3.3) stage.

In [MQ86] and [Serra89], given two sets of parameter estimates at 0 and N, the amplitude and phase angle between the two measurement points are interpolated using the two sets of estimates only. This processing does not guarantee the continuity of the frequency derivative. In [GMMRP03] this is solved by estimating the frequency derivative explicitly at the measurement points. In our work we take a different approach. Since we have used the cubic spline in post-tracking estimation of sinusoidal parameters [WS06] (also see §3.3), which connects frequencies measured from multiple frames with continuous 2nd-order derivative, it is straightforward to apply this to the reconstruction of the sinusoids.

The cubic splines are computed during the re-estimation stage. Between each pair of adjacent measurement points n_l and n_{l+1}, l=1, 2, …, L-1, the frequency is interpolated as a trinomial:

$$\tilde{f}_l^{\circ}(n_l+t)=d+ct+bt^2+at^3 \tag{5. 1a}$$

We integrate (5. 1a) fixing the phase angle of point n_l at $\hat{\varphi}_l$ and get

$$\tilde{\varphi}_l(n_l+n)=\hat{\varphi}_l+2\pi\left(dt+\frac{c}{2}n^2+\frac{b}{3}n^3+\frac{a}{4}n^4\right)+\delta_n \tag{5. 1b}$$

where δ_n, usually a polynomial of n, is the correction term. The choice of δ_n depends on the continuity requirement. For example, to preserve phase and frequency continuity requires

$$\delta_n=\delta_N\frac{n^2}{(\Delta n_l)^2}\left(3-2\frac{n}{\Delta n_l}\right), \tag{5. 2a}$$

and to preserve phase, frequency and frequency derivative continuity requires

$$\delta_n=\delta_N\frac{n^3}{(\Delta n_l)^3}\left(10-15\frac{n}{\Delta n_l}+6\frac{n^2}{(\Delta n_l)^2}\right) \tag{5. 2b}$$

where $\Delta n_l=n_{l+1}-n_l$. The amount of correction is proportional to δ_N. In the minimal correction sense, we find the minimal δ_N that, when applied to (5.1b), satisfies

$$\tilde{\varphi}_l(n_{l+1})=\tilde{\varphi}_{l+1}+2k\pi\text{ , }k\in\mathrm{Z}. \tag{5. 3}$$

The spectral-domain resynthesis method concerning parameters estimated using multiple resolutions is given in Appendix G.

The sinusoids can be subtracted from the original signal to get the residue. In standard sinusoid modeling the sum of sinusoid comprise the *deterministic part* of the audio, while the residue composes the *stochastic part*. The deterministic part is directly composed of individual sinusoid partials, without a mid-level structure; the stochastic part represents the original signal minus the deterministic part [Serra89]. More detailed study of the stochastic part decomposes it into transients and noise [LS98].

5.1.1 Tests

With the resynthesized harmonic sinusoid we are able to evaluate the harmonic sinusoid modeling by the error between the original and resynthesized signals. We run tests on four groups of synthesized signals. The samples are 44100 points long. Amplitude and frequency laws include constant, exponential, and sinusoid-modulated variations. Partial amplitudes are designed to follow a $1/m$ rule, i.e. amplitudes are reciprocal to the partial index. We use the frame size 1024 and hop size 512. The fundamental frequency ranges from 5 bins to 40 bins (1bin=1/1024), spanning 3 octaves. We sample this range every semitone at 37 different pitches. White noises are added to the test sampled optionally. The modeling error is evaluated by a signal-to-noise ratio, where the noise refers to the difference between the original clean signal waveform and the resynthesized harmonic sinusoid. Errors are measured independently for each test sample, then averaged over groups of samples. In these tests post-tracking re-estimation is only performed on the amplitudes using (3.34a).

5.1.1.1 Constant harmonic sinusoids

This group includes 925 test samples, with the 37 fundamental frequencies f^1 from 5bins to 40 bins, 5 stiffness coefficients B from 0 to 0.0008, and 5 signal-to-noise ratios (SNR) from -15dB to 45dB. The phase angles are taken at random. The results are given Table 5.1. For stationary sinusoids the modeling is very successful, with more than 99.9% sinusoid peaks correctly collected into the partials when the SNR is above 15dB. We constantly get slightly better results for higher stiffness coefficients.

This is due to the constraint of *B* above zero, which makes it easier to collect spurious peaks with a positive frequency departure than a negative one.

SNR *B*	-15dB	0dB	15dB	30dB	45dB
0	-0.9	14.8	30.6	45.7	60.7
0.0002	0.3	16.2	32.1	47.2	62.1
0.0004	0.6	16.5	32.4	47.5	62.3
0.0006	0.8	16.8	32.7	47.7	62.6
0.0008	1.0	17.0	32.8	47.9	62.7

Table 5. 1 Resynthesis SNR on constant harmonic sinusoids (dB)

SNR: signal-to-noise ratio; *B*: stiffness coefficient

5.1.1.2 Constant pitch with exponential amplitude

This group includes 1850 test samples, with 37 fundamental frequencies f^1 from 5 bins to 40bins, 2 stiffness coefficients *B* at 0 and 0.0005, 5 amplitude decay rates α at -0.5, -1, -1.5, -2, -2.5 dB/frame (here "per frame" means per hop size, i.e. per 512 points) , and 5 *SNR*s from -15dB to 45dB. The results are given in Table 5.2.

SNR α	-15dB	0dB	15dB	30dB	45dB
-0.5	-0.2	15.2	31.0	46.3	61.2
-1	-0.5	13.9	30.0	45.6	59.7
-1.5	-0.9	12.9	21.7	44.5	56.1
-2	-1.3	11.8	23.7	43.2	49.3
-2.5	-1.8	12.5	21.9	26.7	22.0

Table 5. 2 Resynthesis SNR on harmonic sinusoids with exponential amplitudes (dB)

SNR: signal-to-noise ratio; α: decay rate, in dB/frame

The decay rate has a very regular effect on the error, partially because the signal drops below noise level after certain points. Although in this test all partials have the same decay rate, for partial-dependent decay rates, which is common in real music signals, the behaviour is similar: all partials that falls below the noise level become hard to pick up. Unlike matching pursuits [GB03], sinusoid modeling does not assume any specific coupling between partial amplitudes.

5.1.1.3 Constant pitch with modulated amplitude

This group includes 550 samples, with 22 fundamental frequencies f^1 from 5bins to 40bins (3 octaves on diatonic scale) , 5 modulation depths *d* at 0.1, 0.2, …, 0.5, 5 modulator periods T_M at 2, 4, …, 10 frames, *SNR* is fixed at 15dB. The results are given in Table 5.3. The error increases with modulation depth and frequency.

T_M / *d*	2	4	6	8	10
0.1	28.17	30.34	30.55	30.57	30.60
0.2	24.64	29.57	30.36	30.56	30.56
0.3	21.85	28.60	30.15	30.42	30.49
0.4	19.74	27.54	29.77	30.31	30.44
0.5	18.09	26.58	29.48	30.17	30.39

Table 5. 3 Resynthesis SNR on harmonic sinusoids with modulated amplitudes (dB)

T_M: modulator period, in frames; *d*: modulating depth

5.1.1.4 Pitch modulation with constant amplitudes

This group includes 550 samples, with 22 fundamental frequencies f^1 from 5bins to 40bins (3 octaves on diatonic scale), 5 modulator amplitudes *d* at 0.3, 0.6, …, 1.5 semitones, 5 modulator periods T_M at 2, 4, …, 10 frames, SNR ratio is set to 15dB. We list the resynthesis SNR's in Table 5.4. Only amplitude re-estimation is used in the post-tracking stage to generate these results.

If we compare Table 5.4 with Table 5.3, we see that a frequency modulation of as small as 0.3 semitones brings more error than an amplitude modulation of 50% the central value.

T_M *d*	2	4	6	8	10
0.3	14.5	23.6	27.9	29.0	29.3
0.6	10.8	17.9	21.5	25.4	27.0
0.9	7.7	14.7	17.7	21.3	24.0
1.2	6.0	11.2	13.0	18.5	21.0
1.5	4.8	8.3	7.8	13.0	18.9

Table 5. 4 Resynthesis SNR on harmonic sinusoids with vibrato (dB)

T_M: modulator period, in frames; *d*: modulator amplitude, in semitones

5.2 Application for audio editing

An audio editing operation generates an output audio signal from an input audio signal. It can be as simple as modifying basic audio parameters, such as timing, pitch or loudness, of the input, or as complicated as swapping two musical instruments in a duo. It is common in an audio editor for a user to *select* a part of the whole signal as the target for editing, so that the unselected part remains unmodified. To make it possible, the editor must be able to *decompose* the original signal into the target and non-target parts. The sinusoid model decomposes a sound into deterministic and stochastic parts, therefore enables a user to select the deterministic or stochastic part. It also enables a user to select a sinusoidal partial, since it decomposed the deterministic part into sinusoids. However, the sinusoid model does not provide a straightforward way to access individual events. The harmonic sinusoid model, on the other hand, decomposes the deterministic part into *musical notes* explicitly. With this mechanism, we are able to select a note as the target for editing, and perform standard editing operations on it as if it is an isolated excerpt, without disturbing other audio events.

There are three ways to apply audio editing operations: 1) apply to resynthesized harmonic sinusoids (time-domain post-synthesis editing); 2) apply to sinusoidal parameters (parameter-domain pre-synthesis editing); 3) apply to reconstructed parameter sequences during the resynthesis (in-synthesis editing). Time-domain editing only applies to standard audio editing operations defined for general waveform audio, while parameter-domain and in-synthesis editings apply to a wide range of operations, enabled by the direct access to the model parameters. The in-synthesis editing allows free control of parameters at all samples, including non-measurement points, while the parameter-domain editing only have direct control at the measurement points. In the following we only discuss time- and parameter-domain editings. The in-synthesis editing involves the same operations as the parameter-domain editing, but applies them at every sample instead of at every frame.

Figure 5.1 compares the three types of editing, where the dashed-line box outlines the synthesizer, and the cylinder with "Ed" marks where the modification takes place.

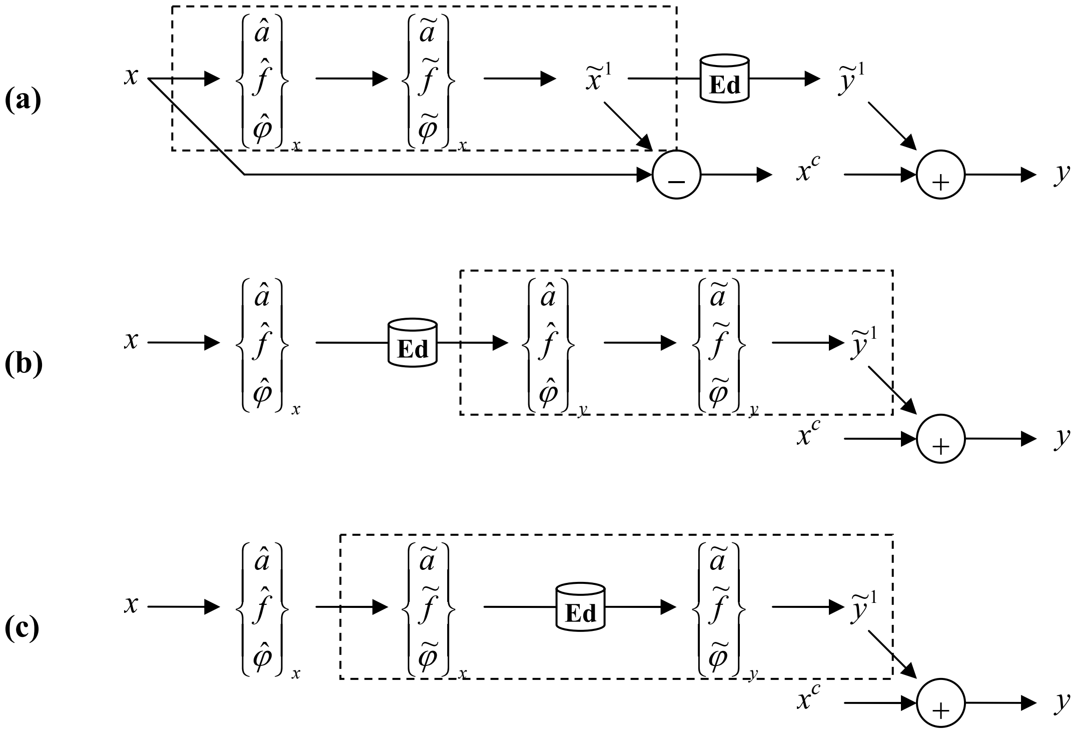


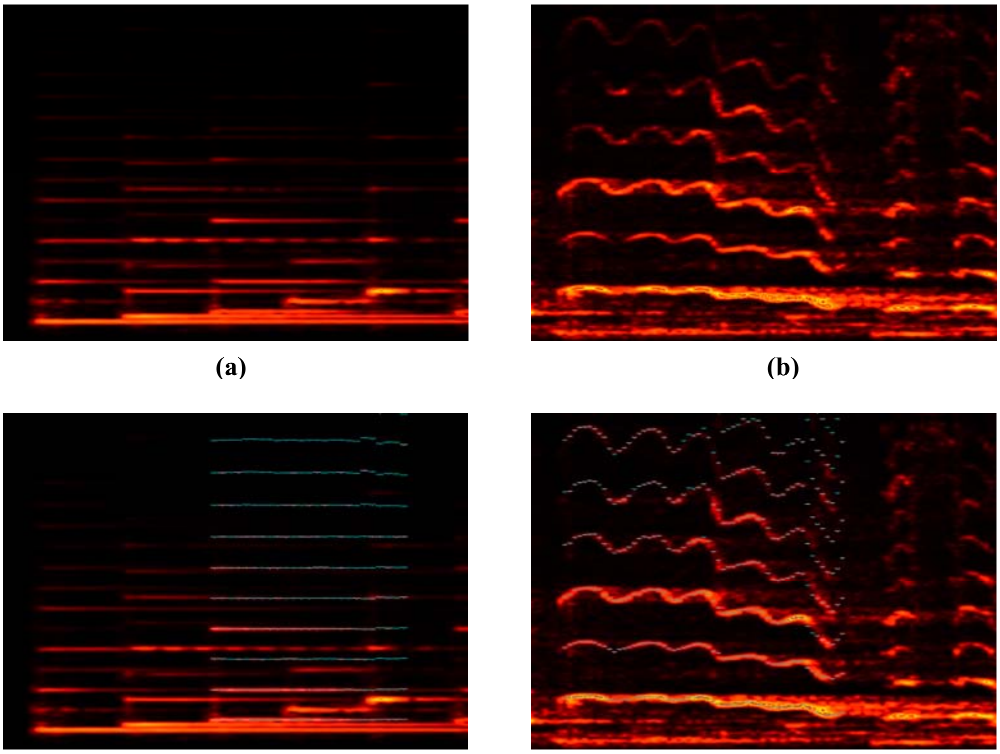
Figure 5. 1 Three ways to apply harmonic sinusoids for audio editing

(a) time-domain; (b) parameter-domain; (c) in-synthesis.

5.2.1 Selecting a target

To perform any audio editing with harmonic sinusoid, the user must be able to specify a target event first. This selection operation must be easy to access, non-ambiguous, and real-time or almost real-time. The spectrogram (Figure 5.2) provides a good image of signal contents with acceptable separation of concurrent events, therefore can serve as the interface for selecting a target event. The simplest way to indicate a target event is by providing a non-ambiguous pair of time and frequency values found on the target, which the harmonic sinusoid tracker uses as the starting point for tracking. In the case that the tracking goes astray, it is convenient to select another pair on the same event. Then the tracking can be forced toward the given direction in the forward-backward framework.

Figure 5.2 (a) is the spectrogram of a solo piano recording; Figure 5.2 (b) is the spectrogram of a solo voice with orchestral accompaniment. The tracking results are shown in Figures 5.2 (c) and (d). In the piano example the third note is selected to be the target, and in the singing example it is the voice. Sound examples are found in \examples directory, named after the figure indices.



(a)

(b)

(c)

(d)

Figure 5. 2 Spectrograms

(a) piano recording; (b) solo voice with accompaniment;
(c)(d) tracking results of the above

5.2.2 The residue

To modify one event within a mixture without changing the other events, we decompose the mixture as an event plus a residue, modify the event, and finally add the modified event back onto the residue. The residue is usually calculated by subtracting the reconstructed event from the mixture, known as subtractive synthesis. Recall the harmonic sinusoids plus noise model

$$x_n = \sum_{k=1}^{K} x_n^k + r_n \tag{5. 4a}$$

where x^k is the k^{th} pitched event, and r is the stochastic part of x. Let $\tilde{x}^1$ be the reconstructed waveform of event x^1. By subtracting $\tilde{x}^1$, we get a residue r^1:

$$r_n^1 = x_n - \tilde{x}_n^1 = x_n^1 - \tilde{x}_n^1 + \sum_{k=2}^{K} x_n^k + r_n \tag{5. 4b}$$

Subtractive synthesis is the only way to ensure perfect reconstruction when no modification is done. However, since the modeling of sinusoids with high dynamics is not perfect, i.e. $\tilde{x}^1$ is only an approximation of x^1, a small error $x_n^1 - \tilde{x}_n^1$ is left in the residue. $x_n^1 - \tilde{x}_n^1$ has the same time-frequency coverage as x^1 itself, which creates a "phantom" echo of the subtracted event. This echo does not necessarily affect the editing quality if the modified event is to be added back to the same position in the time-frequency plane, since the echo will probably be perceptually masked by the modified event [Moore97].

Other methods for calculating the residue include spectral notching and noise modeling. In the spectral notching method a band-stop filter is applied to several bins around the event. This eliminates all signal components close to the removed event, including what is supposed to be left in the residue. The noise modeling method models the stochastic part, i.e. the residue after all harmonic sinusoids have been removed, with a smooth power spectrum. That is, we write

$$x_n = \sum_{k=1}^{K} \tilde{x}_n^k + \sum_{k=1}^{K} (x_n^k - \tilde{x}_n^k) + r_n \tag{5. 4c}$$

The term $\sum_{k=1}^{K} (x_n^k - \tilde{x}_n^k) + r_n$ is a "true" but bad residue containing all the echoes due to inaccurate modeling. The noise modeling technique replace it with a "good" residue $\tilde{r}$, calculated from $x_n - \sum_{k=1}^{K} \tilde{x}_n^k = \sum_{k=1}^{K} (x_n^k - \tilde{x}_n^k) + r_n$ by smoothing away the echoes from the power spectrum. An $\tilde{x}$ can be constructed using $\tilde{r}$ as

$$\tilde{x}_n = \sum_{k=1}^{K} \tilde{x}_n^k + \tilde{r}_n \tag{5. 4d}$$

$\tilde{x}$ is not a perfect reconstruction of x, but they have the same deterministic parts and similar (in the power spectrum sense) stochastic parts, so they will sound similar. The residue for $\tilde{x}^1$ is generated as

$$\tilde{r}_n^1 = \tilde{x}_n - \tilde{x}_n^1 = \sum_{k=2}^{K} \tilde{x}_n^k + \tilde{r}_n \tag{5. 4e}$$

$\tilde{r}_n^1$ does not carry an echo of x^1.

Figure 5.3 shows the resynthesized harmonic sinusoids and corresponding subtractive residues for the two examples above. Clean subtraction is achieved for the piano example, while a “phantom” echo is left from the vocal excerpt. This highlights the necessity of parameter re-estimation for varying-frequency sinusoids. Audio samples of these are found in /examples.

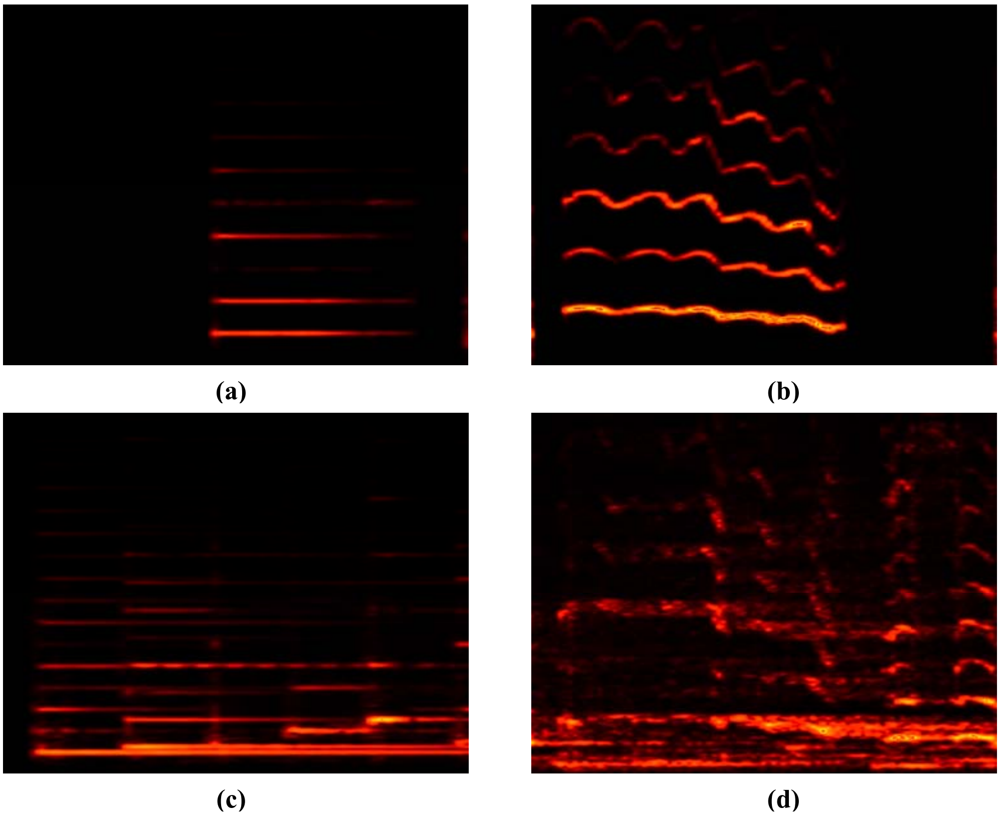


Figure 5. 3 Extracted event and residue

(a)(b) resynthesized harmonic sinusoids; (c)(d) residues calculated by subtraction

In the following we denote the input signal x, the reconstructed event $\tilde{x}^1$, the residue x^c, the modified event $\tilde{y}^1$, and the output signal y.

5.2.3 Cut, copy and paste

These operations are extremely common in editors that handle *objects*. As the harmonic sinusoid modeling represents a pitched event as an object, it is possible to cut, copy and paste it in an audio editor. The cut operation is implemented as

$$y_n = x_n^c \,,\ \tilde{x}^1 \rightarrow clipboard \,, \tag{5. 5a}$$

and paste is implemented as

$$\tilde{x}^1 \leftarrow clipboard \,,\ y_n = x_n + \tilde{x}_n^1 \tag{5. 5b}$$

It is common to do time-shifting by combining the cut and paste operations:

$$y_n = x_n^c + \tilde{x}_{n-k}^1 \tag{5. 5c}$$

where k is the time shift, in samples.

5.2.4 Amplification

Time-domain amplification operation is performed as

$$\tilde{y}_n^1 = A_n \tilde{x}_n^1 \tag{5. 6a}$$

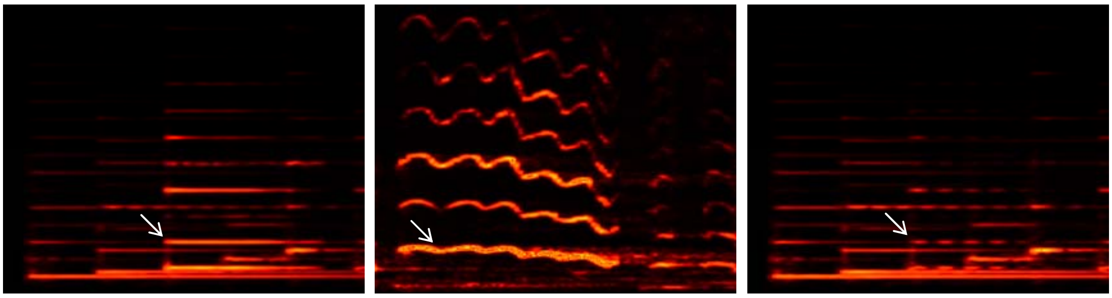
where A_n is a *gain* factor for point n. The parameter-domain version is

$$\left(a_l^m\right)_y = A_{n_l}\left(\hat{a}_l^m\right)_x \,,\ \forall\, l,\, m \tag{5 .6b}$$

Pure amplification is specified by setting A_n (or A_{n_l}) as a constant. Using pure amplification it is possible to re-balance between notes or voices, or the same note in difference channels. These are basic operations in the remixing of already mixed audio.

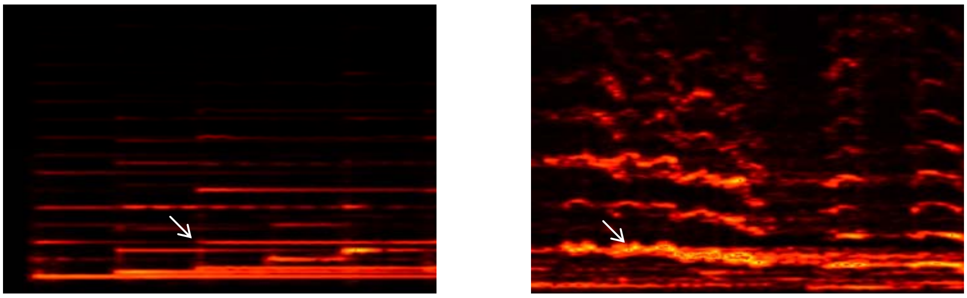
Time-variant amplification uses non-constant gain factors, known as *envelopes*. Typical envelopes include rectangular windows for trimming an event, fade-in and fade-out windows, and sinusoid or other periodic windows for amplitude modulation. Unlike the time-domain method, the parameter-domain method does not have direct access on the gain factors between measurement points, which are implicitly interpolated after the modification. Since slow variation is assumed during resynthesis, in parameter domain we cannot implement fast-varying amplitude envelope.

Figure 5.4 gives several examples of amplification. In Figures 5.4 (a) and (b) the two harmonic sinusoids are amplified by 6dB. Figure 5.4 (c) illustrates the artificial amplitude modulation on a selected event. These examples are also found in audio format in \examples.



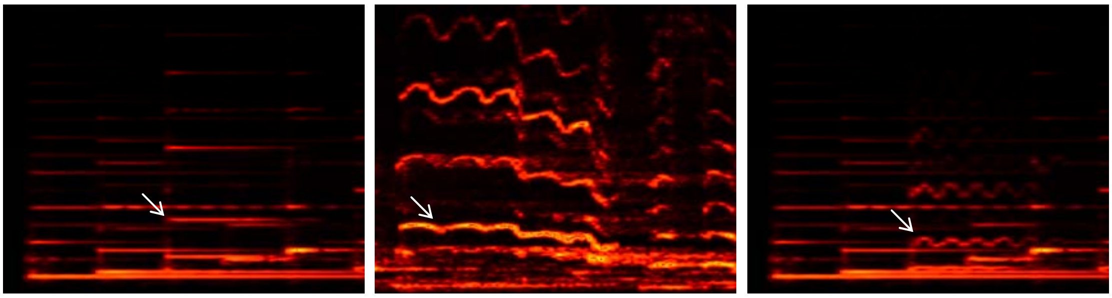
(a) **(b)** **(c)**

Figure 5. 4 Amplification on selected event
(a)(b) constant amplification; (c) amplitude modulation



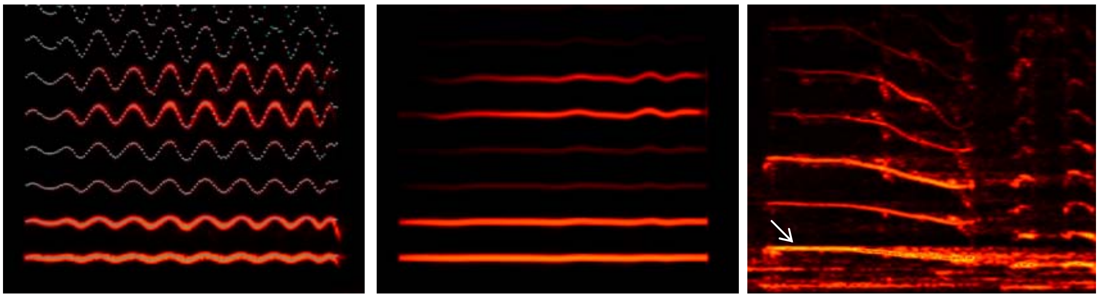
(a) **(b)**

Figure 5. 5 Time stretching
(a) α=2; (b) α=0.6



(a) **(b)** **(c)**

Figure 5. 6 Pitch shifting
(a) up by a triton; (b) up by a fifth; (c) pitch modulation



(a) **(b)** **(c)**

Figure 5. 7 Frequency de-modulation
(a) a vibrato; (b) de-modulated vibrato; (c) de-modulated glissando

5.2.5. Filtering

Time-domain filtering is performed as

$$\tilde{y}^1 = \tilde{x}^1 * h \text{, or } \tilde{y}^1_n = \sum_m \tilde{x}^1_m h_{n-m} \tag{5. 7a}$$

where h is the response of the filter. Time-variant filtering is implemented by using time-dependent response, say h_n:

$$\tilde{y}^1_n = \sum_m \tilde{x}^1_m h_{m,n-m} \tag{5. 7b}$$

The filter response h is determined from specifications on frequency response using some filter design procedures. Let the desired frequency response be $H(f)$, parameter-domain filtering is implemented as frequency-dependent amplification:

$$\left(\tilde{a}^m_l\right)_y = H(\hat{f}^m_l)\cdot\left(\hat{a}^m_l\right)_x,\ \forall\, l, m \tag{5. 7c}$$

Time-variant filtering is implemented using frame-dependent frequency response:

$$\left(\tilde{a}^m_l\right)_y = H_{n_l}(\hat{f}^m_l)\cdot\left(\hat{a}^m_l\right)_x,\ \forall\, l, m \tag{5. 7d}$$

While the frequency response is frame-dependent, the synthesizer smoothly interpolates it from one frame to the next.

5.2.6. Time stretching

Time-domain time stretching is implemented on $\tilde{x}^1$ with standard time-stretching methods [Portnoff81, LD99]. In parameter domain time stretching is performed on the *measurement time* parameters:

$$\left(n_l\right)_y = n_c + \alpha\left(\left(n_l\right)_x - n_c\right),\ \forall\, l \tag{5. 8}$$

where n_c is a position chosen to have the same timing in x and y, i.e. the fixed point, and α is the stretching rate. When α>1 the harmonic sinusoid is stretched; when 0<α<1 it is compressed; when α<0 it is reversed with stretching rate -α.

Figure 5.5 shows time stretching. In (a) the third piano note is stretched by 100%, while in (b) the selected vocal part is compressed by 40%. The audio examples are found in \examples.

5.2.7. Pitch shifting

From the time-scale modification point of view, a pitch shifting given the shifting rate α (new pitch divided by the original pitch) is equivalent to time stretching of stretching rate α coupled with resampling of resampling rate α. Time-domain pitch shifting is implemented on $\tilde{x}^1$ with standard pitch-shifting methods [Portnoff81]. In the parameter domain, pitch shifting is performed on individual partial frequencies:

$$\tilde{f}_l^m = 2^{\beta/12} \cdot \hat{f}_l^m, \ \forall l, m \qquad (5.\ 9a)$$

where β=12·log$_2\alpha$ is the amount the shifting in semitones. When β>0 the pitch is shifted up; when β<0 it is shifted down. An anti-alias filter should be applied in combination with upward pitch-shifting:

$$H(f) = \begin{cases} 1, & f < 0.45 \\ (0.5 - f)/0.05, & 0.45 \le f \le 0.5 \\ 0, & f > 0.5 \end{cases} \qquad (5.\ 9b)$$

Frequency modulation is implemented using pitch shifting with a frame-dependent amount of shift.

The pitch shifting in (5.9a) moves the formants along with the partials. Since the positions of the formants characterise the shape and size of a resonant body, shifting the formants along may alter the resonant characteristics, and therefore affect the perception of the sound source.

Figure 5.6 gives several examples of pitch shifting. In (a) the third piano note is shifted up by 6 semitones (a “triton”); in (b) the voice is shifted up by 5 semitones. Figure 5.6 (c) illustrates the artificial pitch modulation on the selected note. These examples are also found in audio format in \examples.

5.2.8. Amplitude and frequency demodulation

Demodulation is the inverse process of modulation. While in modulation a modulated signal is generated from a carrier and a modulator, in the reverse process a carrier and a modulator is separated from a modulated signal.

In real-world music, amplitude modulation may appear by itself, while frequency modulation is almost always accompanied by amplitude modulation (see 1.2.3). Here we consider these two cases.

5.2.8.1. Pure amplitude de-modulation

For the m^{th} partial, let the estimates of a modulated amplitude be $\hat{a}^m$, the carrier amplitude be $\bar{a}^m$, and the modulator be $\underline{a}^m$, so that

$$\hat{a}_l^m = \bar{a}_l^m + \underline{a}_l^m, \ \forall\, l, m \tag{5. 10a}$$

We always assume that the carrier varies slower than the modulator, to estimate $\bar{a}^m$ and $\underline{a}^m$ we decompose $\hat{a}^m$ into a slow-varying part and a fast-varying part, assign the previous to $\bar{a}^m$, and the latter to $\underline{a}^m$. It is up to the user to decide where to draw the line between "slow" and "fast". For example, if we find the slow-varying part by the following moving average:

$$\bar{a}_l^m = \frac{\sum_{k=-K}^{K} w(k/K)\hat{a}_{l+k}^m}{\sum_{k=-K}^{K} w(k/K)} \tag{5. 10b}$$

where w is a low-pass window function defined on [-1, 1], then the window width $2K$ controls the balance between slow and fast. The larger is K, the slower-varying is $\bar{a}_l^m$.

The parameter-domain amplitude de-modulation is given as

$$\left(\hat{a}_l^m\right)_y = \left(\bar{a}_l^m\right)_x, \ \forall\, l, m \tag{5. 11a}$$

and the re-modulation is given as

$$\left(\hat{a}_l^m\right)_y = \left(\bar{a}_l^m\right)_x + Ed[\left(\underline{a}_l^m\right)_x] \tag{5. 11b}$$

where Ed[·] represents a general operator. For example, Ed[·] being an amplifier changes the modulating depth, and Ed[·] being a time stretcher changes the modulating rate.

Amplitude de-modulation is difficult in time-domain processing unless all partials of $\tilde{x}^1$ are modulated with the same modulator phase and the same modulating depth.

5.2.8.2 Frequency de-modulation

The frequency estimate $\hat{f}^m$ can be decomposed into a carrier $\bar{f}^m$ and a modulator $\underline{f}^m$ in the same way as the amplitude estimate. The parameter-domain frequency de-modulation is given as

$$\left(\hat{f}_l^m\right)_y = \left(\bar{f}_l^m\right)_x \tag{5. 12a}$$

and the re-modulation is given as

$$\left(\hat{f}_l^m\right)_y = \left(\bar{f}_l^m\right)_x + Ed[\left(\underline{f}_l^m\right)_x] \tag{5. 12b}$$

However, frequency modulation rarely comes without accompanying amplitude modulation. When the frequency estimates are de-modulated, so should the accompanying amplitude modulation. However, the frequency-modulation-related amplitude modulation is a complicated phenomenon out of the scope of this thesis. Here we only propose a method based on the simplified source-filter model given in (1.19a):

$$a_l^m = A_m A_s(n_l) A_f(f^m(n_l)) \equiv \overline{A}_l\ A^m(f^m(n_l)) \tag{5. 13a}$$

where $\overline{A}$, defined as $\overline{A}_l = A_s(n_l)$, is a slow-varying amplitude carrier, and $A^m(f)$ combines the source model A_m and filter model A_f. $\overline{A}$ is estimated as the slow-varying part of the overall amplitude. For every partial m, A^m is expressed as a function of f by mapping the frequency and amplitude estimate pairs $\{\left(\hat{f}_l^m\right)_x, \left(\hat{a}_l^m\right)_x / \overline{A}_l\}_l$ onto a f-A plane, then fit the points with a smooth curve. The de-modulation of amplitudes is then given as

$$\left(\hat{a}_l^m\right)_y = \overline{A}_l\ A_f^m\left(\left(\hat{f}_l^m\right)_y\right) \tag{5. 13b}$$

In other words, $\left(\hat{a}_l^m\right)_y$ (divided by $\overline{A}$) is predicted as a median of those amplitude estimates (divided by $\overline{A}$) which have the same partial index and are closest in frequency to $\left(\hat{f}_l^m\right)_y$.

Pure frequency de-modulation is accomplishable in time-domain processing as pitch-dependent pitch shifting. Frequency de-modulation coupled with amplitude de-modulation is difficult in time domain.

Figure 5.7 illustrates the joint amplitude-frequency demodulation. Figure 5.7 (a) is the spectrogram of a vocal vibrato, with the harmonic tracking results. Figure 5.7 (b) gives the de-modulated vibrato, which shows high stability in articulation. In Figure 5.7 (c) we apply the demodulation on the accompanied vocal excerpt, in which case the vibrato accompanying a vocal glissando is removed. These excerpts are also found as audio files in \examples.

5.2.9. Section summary

In §5.2 we have discussed the applications of harmonic sinusoids for audio editing. The examples given in this section are only basic operations. There are many other possibilities of applying harmonic sinusoid model for audio editing, such as adding and removing partials, imposing and removing formants, swapping parameters between events, etc. Some of these are not possible without harmonic sinusoid modeling.

5.3 Other applications

Apart from the above, the harmonic sinusoid modeling also enables a wide range of other applications, some of which are listed in this section.

5.3.1 Instrument / singer recognition

Most music we encounter in daily life are polyphonic, in which multiple pitched music notes are allowed at the same time. The polyphony adds to the difficulty of some music information retrieval tasks which are designed to extract information regarding individual musical notes, or a monophonic excerpt. Examples of such tasks include pitch identification and musical instrument / singer recognition. When the monophonic methods are applied to polyphonic, it is either that the result be given an alternative interpretation, such as *predominant pitch* instead of plain pitch [KVH00], or some specific method be introduced to combat the influence of the unwanted contents [BE01, TWRCY03]. The harmonic sinusoid, on the other hand, directly represents a single music note within an audio mixture, so a monophonic retrieval method can be directly applied to it without considering other components. A very similar idea has been explored in [FKGKOO05] for singer identification, with promising results. Standard recognizers use spectral features such as the linear prediction cepstrum coefficient (LPCC) [MG76], or Mel-frequency cepstrum coefficients (MFCC) [Logan00], most of which can be calculated directly from the sinusoidal parameters instead of from resynthesized audio.

5.3.2 Advanced audio annotation

In music information retrieval tasks it is crucial to have a properly annotated database. The type of annotation used is determined by specific retrieval tasks, e.g. onsets positions are annotated for onset detection, pitch values and duration are annotated for transcription, etc. Most currently available annotations fall into a *when-and-what* style, i.e. something happens at some time. With such an annotation, a label identifying "what" is tagged onto a point or a segment on the time axis, which maps onto the audio content.

One problem of this kind of annotation is the polyphony, i.e. concurrent events. In the current method all events that happen at the same time are tagged to the same position. Accordingly, when one is trying to locate some event A, a tag is associated with the duration of A only. Using this information one may access an audio excerpt containing the event A, but probably not only A. In other words, one has direct access only to the duration of an event, but not to the event itself.

With the help of harmonic sinusoid model, it is possible to annotate a pitched event as a harmonic sinusoid. In this case the label is tagged onto a time-varying track in the time-pitch plane instead of an interval on the time axis alone. The additional dimension makes it possible to discriminate between annotated concurrent events, and to refer to one event among concurrent events at a higher SNR. Even in the monophonic case, the harmonic sinusoid modeling is useful in providing detailed annotations of time-varying parameters, such as the instantaneous pitch within a vibrato.

5.3.3 Weak onset detection

Music onset detection has been extensively addressed in [BDADDS05] and [Dixon06]. In these works the onsets are detected according to variations of audio properties. This idea, however, does not work for events with slow starts (fade-in), since they lack obvious variations. For the same reason, offset detection is difficult for the onset detection methods, since many offsets happen as fade-outs.

The problem is that few onset detection methods consider the relation between onsets and events, i.e. an onset exists if and only if there is an event it can be attached to. In [WS05] we proposed a method for discarding spurious onsets when no new events can be detected. This can also be carried out inversely: if there is an event A detected at time t_1, but A is not detectable at time $t_0<t_1$, then there exists an onset of A between t_0 and t_1. Since the harmonic sinusoid model explicitly models pitched events, they can be used to detect onsets, especially those with slow starts. There can still be problem of locating a weak onset accurately in time. However, this can be hard for a human annotator too.

5.3.4 Music coding

The harmonic sinusoid model is compatible the MPEG-4 “Harmonic and Individual Lines plus Noise” coding tool [PEF98, PM00], with extra structures such as the inharmonicity, and models smooth parameter adaptation between frames.

5.3.5 Analysis of music performance

The harmonic sinusoidal parameters reflect several technical aspects of music performance, including pitch contour, dynamics, damping, tremolos and vibratos, timbre control, etc. Musicologists can use these details to study the specific articulations or to compare different performances. It is also possible to automatically monitor training sessions for educational purposes.

5.3.6 New audio features

The harmonic sinusoid provides a moderately compact representation which effectively preserves the information of a pitched sound. Features can be extracted from sinusoidal parameters to compare the similarity between harmonic sinusoids. Some instruments are known to have distinct characteristics in the harmonic sinusoid representation, such as missing partials, which are not effectively represented by other features.

5.4 Summary

In this chapter we have discussed several applications of harmonic sinusoid models, including audio editing, musical instrument recognition, etc. These applications are enabled by harmonic sinusoid modeling in two ways: the extraction of pitched events from other events, and the quantification of harmonic sinusoidal parameters. In short, the harmonic sinusoid model offers an inspection into the harmonic structure of individual events, and functions as an interface between waveform audio and note-level representations of music.

Chapter 6

Conclusion and perspectives

This chapter summarizes the work presented in this thesis, and highlights its contributions to the research. In the perspective section we summarize several weak points of the current system, and propose how the modeling can be further improved.

6.1 Thesis summary

This thesis presents the theory, implementation, and applications of the harmonic sinusoid model. The harmonic sinusoid is introduced to represent the deterministic component of a pitched event in the sound. Unlike the standard sinusoid model, which describes the deterministic audio components in a general sense without distinguishing their sources, the harmonic sinusoid model explicitly represents the deterministic parts of individual sound sources (or multiple sources in unison), therefore provides a note-level representation of music audio.

Chapter 1 defines the definition of the model directly in 1.1, and examples of harmonic sinusoids are given in 1.2 with music acoustical contexts. We have also discussed several aspects of the time-varying sinusoid, including the uniqueness and slowness issues, and its behaviour in the frequency domain.

Chapter 2 reviews the techniques used in standard sinusoid modeling, including peak picking, parameter estimation, sinusoid tracking and resynthesis methods. In addition to a brief overview, for peak picking and frequency estimation we have explored the methods analytically to show why, and how well, they work. For resynthesis we reformulated the standard trinomial frequency-phase interpolation in a interpolation-plus-correction frame work, which leads to the finer resynthesis method to be given in Chapter 5. After reviewing the standard sinusoid model we show how this model can be upgraded to the harmonic sinusoid model, give the outline of the harmonic sinusoid modeling system, and draw a comparison with the standard sinusoid modeling system.

Chapter 3 presents methods for measuring harmonic sinusoidal parameters. The LSE estimator is presented in 3.1, with emphasis on its frequency-domain implementation, and a discussion of its behaviour on time-varying sinusoids. 3.2 discusses harmonic grouping, i.e. grouping sinusoid atoms into harmonic particles according to partial harmonicity. The core technique in harmonic grouping is a robust inequality-based representation of harmonic frequencies, which tolerates missing partials, frequency estimation errors, as well as selected type of inharmonicity. Under this model a group of harmonic frequencies is represented as an area in some fundamental-inharmonicity plane. Several aspects of this model are discussed in 3.2.3~3.2.6. In 3.2.7 we finally present the harmonic grouping techniques based on the inequality model. 3.3 and 3.4 are devoted to post-tracking estimation of time-varying sinusoids. Two independent algorithms are presented to show how the knowledge on parameter variation can be used for estimating sinusoidal parameters at higher accuracy.

Chapter 4 discusses harmonic sinusoid tracking. The inequality-based harmonic frequency representation is used for providing frequency evolution boundaries to ensure the continuity of inharmonicity property. Frequency and amplitude continuity criteria are discussed in 4.1 and 4.2, respectively. 4.3 discusses forward tracking of a single harmonic sinusoid, based on the continuity criteria. The forward tracking is performed jointly with harmonic grouping to ensure harmonic particles being optimal in the continuity sense. 4.4 discusses the extension of this tracking scheme onto multiple harmonic sinusoids, which combines the method for grouping multiple harmonic particles in Chapter 3, and the forward tracking method in 4.3. Finally we discussed the application of forward-backward method in harmonic sinusoid tracking, which shows better performance in a noisy environment.

Chapter 5 presents applications of the harmonic sinusoid model. We first discuss the harmonic sinusoid synthesizer to conclude the analyzer-synthesizer modeling cycle. The emphasis of the application part is put on audio editing. The harmonic sinusoid model enables two new features for audio editing: object-based editing allows a user to modify a selected pitched event with little effect on other events, and the direct access to sinusoidal parameters provides possibilities for new sound effects. Several other applications are briefly proposed as suggestions for future investigation.

6.2 Contributions

The main contributions of this thesis are listed as follows.

- Harmonic sinusoid modeling system (2.6). This thesis formulates harmonic sinusoid modeling as an analysis and synthesis system based on harmonic particles, which can be regarded as an upgrade from standard sinusoid modeling in a similar framework, but with distinct internal structures and algorithms. A harmonic sinusoid directly models the deterministic part of a pitched audio event, therefore have a wider spectrum of applications than the individual sinusoids in a standard sinusoid model.

- Robust representation of harmonic frequency contents (3.2.3). The inequality-based representation of harmonic frequencies provides a way for combating several issues that corrupt the harmonic grouping of sinusoid atoms, including the absence of partials, inaccurate frequency estimates, and inharmonicity.

- Method for finding harmonic signal components (harmonic particles) from the spectrum (3.2.7). The harmonic grouping method based on the above harmonic frequency representation is capable of finding harmonic particles against noise, weak or masked partials, frequency estimation errors, and inharmonicity. A method for finding concurrent harmonic particles considering shared spectral peaks is also proposed.

- Continuity criteria for tracking harmonic partials (4.1 and 4.2). Unlike the standard sinusoid modeling, which tracks sinusoids using frequency continuity only, in harmonic sinusoid tracking we use a combination of continuity criteria concerning both amplitude and frequency, and even higher-level clues such as the amplitude distribution among partials.

- Joint operation of harmonic grouping and harmonic tracking (4.3~4.6). Unlike the standard sinusoid modeling, which tracks sinusoids by connecting pre-detected sinusoid atoms, in harmonic sinusoid track we propose to perform harmonic grouping jointly with the tracking process to guarantee inharmonicity continuity and tracking criteria optimization. This joint operation can be applied in both the forward and the forward-backward tracking schemes.

- Methods for estimating sinusoids using the knowledge of signal dynamics (3.3). The accurate estimation of the parameters of time-varying sinusoids remains a bottleneck of accurate sinusoid analysis and synthesis. However, after the tracking stage, it is possible to re-estimate the parameters using the information embedded in the tracks. For the specific case of the LSE estimator, we examine its behaviour when applied to time-varying sinusoids to show how the parameter variation affects the estimation accuracy, then propose a methods to correct the error. In a wider sense, we have proposed a de-variation method that works with a general sinusoid estimator (Appendix E.3).

- Method for choosing the window size according to signal dynamics (3.4). As signal dynamics are better captured by shorter windows, we have proposed a method for automatically selecting a shorter window size when the signal dynamics is too high for the current window. This method is less accurate than the above re-estimation method but is able to work with higher signal dynamics than the above method can handle.

- Application of harmonic sinusoids for audio editing (5.2). This is probably the most straightforward application of the harmonic sinusoid modeling. We have shown that with harmonic sinusoids, we can achieve standard audio editing operation on individual pitched sounds within a mixture, or invent new sound effects not easily available with conventional methods.

Minor contributions include

- Discussions on time-varying sinusoids, which, although plays a central role in standard sinusoid modeling, has been largely ignored (1.3, 1.4). The discussion on the Fourier transform of time-varying sinusoids leads to both post-tracking parameter estimation methods.

- An analytical review of DFT-based frequency estimation methods (2.2).
- Spectral-domain resynthesis of time-varying sinusoids with multiple resolutions (Appendix F).
- An application of harmonic particles for audio transcription [WS05b].
- Proposals of applications of harmonic sinusoids (5.3).

6.3 Perspectives

The current harmonic sinusoid modeling techniques can be further improved on several aspects, listed as follows.

- The partial harmonicity has its origin in one-dimension simple harmonic oscillation of a string and an air column, as well as periodically-stimulated resonant bodies. It, however, does not describe membrane or bar vibration, which lies behind many percussion instruments such as the semi-pitched kettledrum, or the pitched marimba. The sinusoid modeling of these instruments requires partial frequency coupling rules different from simple harmonicity.
- Even for harmonic instruments, there may exist extra partials that do not fall within a harmonic context, known as *phantom partials* [Conklin99]. They are either generated from multiple vibrating bodies, such as unison strings in a piano, or from non-linear coupling between vibrating modes. These can be picked up by introducing individual spectral lines into the model, or be included in a more comprehensive harmonic model.
- Many pitched musical events involve transients, especially at the onsets. The harmonic sinusoid model does not come with transients, therefore is incapable of representing these events in a perceptually complete form. It is desirable to develop transient detection and representation methods that can associate transients with corresponding harmonic sinusoids and draw an end to this incompleteness.

- Harmonic tracking can be further refined by introducing finer frequency and amplitude continuity criteria, exploring the use of forward-backward tracking with beam trimming, or tracking harmonic sinusoids using sound source profiles. On the other hand, since the harmonic sinusoids represent note-level events, which are directly perceivable by human, it is also interesting to develop human-aided tracking algorithms to boost performance.
- The current harmonic sinusoid modeling treats very close (or overlapping) partials from two or more harmonic sinusoids as a shared partial. Although the distribution of a shared partial has been discussed in the maximal-continuity sense in (4.14c), there is no guarantee that it is an optimal solution. Better techniques to resolve shared partials may require further investigations.
- On the synthesizer side, a more accurate modeling of time-varying sinusoids is necessary for obtaining cleaner residues. The current re-estimation method can at the best achieve an optimal *frame-level* approximation of an arbitrary sinusoid track, but has little capacity on the sub-frame level parameter variations. A better method will probably work on a smaller scale than frames, or even operate on sample-level, to gain full control of parameter variations.

Finally, a wide variety of applications of harmonic sinusoids has been left unexplored. Some possible applications have been suggested in 5.3. However, we believe there are always other possibilities beyond our imagination, and it is crucial to work with potential users of this technique to find new applications, which in turn will raise additional issues for further developing the techniques. The harmonic sinusoid model builds a direct connection between waveform audio and musical notes, through which now it is time for each side to venture into the other.

Bibliography

[AF95] F. Auger and P. Flandrin, “Improving the readibility of time-frequency and time-scale representations by the reassignment method,” *IEEE Tran. Signal Proc.*, vol.43 no.5, 1995, pp.1068-1089.

[AKZ99] R. Althoff, F. Keiler and U. Zölzer, “Extracting sinusoids from harmonic signals,” in *Proc. DAFx*’99, Trondheim, 1999.

[AR77] J. B. Allen and L. R. Rabiner, “A unified approach to short-time Fourier analysis and synthesis,” in *Proceedings of the IEEE*, vol.65 no.11, Nov. 1977, pp.1558-1564.

[ASA60] American Standards Association. *Acoustical Terminology*. 1960.

[ASP91] S. Austin, R. Schwartz, P. Placeway, “The forward-backward search algorithm,” in *Proc. ICASSP91*, Toronto, 1991.

[BC94] G. J. Brown and M. Cooke, “Computational auditory scene analysis,” Computer Speech and Language, 8, 1994, pp. 297-336.

[BDADDS05] J. P. Bello, L. Daudet, S. Abdallah, C. Duxbury, M. Davies and M.B. Sandler, “A tutorial on onset detection in music signals,” *IEEE Trans. Speech and Audio Proc.*, vol. 13, no.5, 2005, pp. 1035-1047.

[BE01] A. Berenzweig and D. P. W. Ellis, “Locating singing voice segments within music signals,” in *Proc. WASPAA’01*, New Paltz, 2001.

[BP93] J. C. Brown and M. S. Puckette, “A high resolution fundamental frequency determination based on phase changes of the Fourier transform,” *J. Acous. Soc. Am.*, vol.94, no.2, 1993, pp.662-667.

[Brown92] J. C. Brown, "Musical fundamental frequency tracking using a pattern recognition method," *J. Acous. Soc. Am.*, vol.92 no.2, 1993, pp.1394-1402,

[Carlson81] A. B. Carlson, *Communication systems*, 2nd edition. McGraw-Hill Inc., 1981.

[Carson22] J. R. Carson, "Notes on the theory of modulation," *Proceedings of the IRE*, vol.10 no.1, 1922, pp.57-64.

[CD07] P. Cunningham and S. J. Delany, k-Nearest neighbour classifiers. Technical report UCD-CSI-2007-4. University College, Dublin. 2007.

[Conklin99] H. A. Conklin, "Generation of partials due to nonlinear mixing in a stringed instrument", *J. Acoust. Soc. Am.*, vol.105, no.1, January 1999, pp.536-545.

[DD07] C. Dubois and M. Davy, "Joint detection and tracking of time-varying harmonic components: a flexible Bayesian approach," *IEEE Tran. Au. Spe. Lang. Proc.*, vol.15 no.4, 2007, pp.1283-1295.

[DG02] M. Davy and S. J. Godsill, "Bayesian harmonic models for musical pitch estimation and analysis", in *Proc. ICASSP'02*, Orlando, 2002.

[DG03] M. Davy and S. J. Godsill, "Bayesian harmonic models for musical signal analysis," *Bayesian Statistics*, 7, 2003.

[DGR93] P. Depalle, G. Garcia and X Rodet, "Tracking of partials for additive sound synthesis using Hidden Markov Models," in *Proc. ICASSP'93*, Minneapolis. 1993.

[Dixon06] S. Dixon, "Onset detection revisited," in *Proc. DAFx'06*, Montreal, 2006.

[DM00] M. Desainte-Catherine and S. Marchand, "High precision Fourier analysis of sounds using signal derivatives," *J. AES*, vol.48 no.7/8, 2000, pp.654-667.

[Ellis96] D. P. W. Ellis, *Prediction-driven computational auditory scene analysis*, PhD thesis, MIT, 1996.

[EK00] A. Eronen and A. Klapuri, “Musical instrument recognition using cepstral coefficients and temporal features”, in *Proc. ICASSP’00*, Istanbul, 2000.

[ERD05] S. Essid, G. Richard and B. David, “Instrument recognition in polyphonic music,” in *Proc. ICASSP’05*, Philadelphia, 2005.

[FKGKOO05] H. Fujihara, T. Kitahara, M. Goto, K. Komatani, T. Ogata, and H. G. Okuno, “Singer identification based on accompaniment sound reduction and reliable frame selection,” in *Proc. ISMIR05*, pp. 329–336, London, 2005.

[FR98] N. H. Fletcher and T. D. Rossing, *The Physics of Musical Instruments* (2nd Ed.), Springer-Verlag New York, Inc., 1998.

[GB03] R. Gribonval and E. Bacry, “Harmonic decomposition of audio signals with matching pursuit,” *IEEE Tran. Signal Proc.*, vol.51 no.1, 2003.

[GD05] S. Godsill and M. Davy, “Bayesian computational models for inharmonicity in musical instruments,” in *Proc. WASPAA’05*, New Paltz, 2005.

[GHNO02] M. Goto, H. Hashiguchi, T. Nishimura and R. Oka, “RWC music database: Popular, classical, and jazz music databases,” in *Proc. ISMIR’02*, Paris, 2002, pp.287-288.

[GMMRP03] L. Girin, S. Marchand, J. di Martino, A. Röbel and G Peeters, “Comparing the order of a polynomial phase model for the synthesis of quasi-harmonic audio signals,” in *Proc. WASPAA’03*, New Paltz, 2003, pp.193-196.

[Goto04] M. Goto, “A real-time music-scene-description system: predominant-F0 estimation for detecting melody and bass lines in a real-world audio signals,” *Speech Communication*, vol.43, 2002, pp.311-329.

[Hartmann96] W. M. Hartmann, “Pitch, periodicity, and auditory organization,” *J. Acoust. Soc. Am.*, vol. 100, 1996, pp. 3491–3502.

[Howe75] H. S. Howe Jr., *Electronic Music Synthesis: Concepts, Facilities, Techniques*, W. W. Norton & Company, 1975.

[Kay87] S. M. Kay, *Modern spectral estimation: theory and application*, Prentice Hall, Englewood Cliffs, 1987.

[Klapuri99] A. Klapuri, "Wide-band pitch estimation for natural sound sources with inharmonicities," in *Proc. AES 106th Convention*, Munich, 1999.

[Klapuri01] A. Klapuri, "Multipitch estimation and sound separation by the spectral smoothness principle," in *Proc. ICASSP'01*, Salt Lake City, 2001.

[KM02] F. Keiler and S. Marchand, "Survey on extraction of sinusoids in stationary sounds," in *Proc. DAFx'02*, Hamburg, 2002, pp.51-58.

[KVH00] A. Klapuri, T. Virtanen and J.-M. Holm, "Robust multipitch estimation for the analysis and manipulation of polyphonic musical signals," in *Proc. DAFx'00*, Verona, 2000.

[LD99] J. Laroche and M. Dolson, "New phase-vocoder techniques for pitch-shifting, harmonizing and other exotic effects," in *Proc. WASPAA'99*, New Paltz, 1999.

[LMRR03] M. Lagrange, S. Marchand, M. Raspaud and J.-B. Rault, "Enhanced partial tracking using linear prediction," in *Proc. DAFx'03*, London, 2003.

[Logan00] B. Logan, "Mel frequency cepstral coefficients for music modeling," in *Proc. ISMIR'00*, Plymouth, Massachusetts, 2000.

[LS98] S. N. Levine and J. O. Smith III, "A sines+transients+noise audio representation for data compression and time/pitch scale modifications," *in Proc. AES 105th Convention*, San Francisco, 1998.

[Mallat99] S. Mallat, *A Wavelet Tour of Signal Processing* (2nd Ed.), Academic Press, 1999.

[Métois98] E. Métois, "Musical gestures and audio effects processing," in *Proc. DAFx'98*, Barcelona, 1998.

[MG76] J. D. Markel and A. H. Gray, Linear prediction of speech, New York: Springer-Verlag, 1976.

[Moore97] B. C. J. Moore, *An introduction to the psychology of hearing*, Academic Press, 1997.

[MQ84] R. J. McAulay and T. F. Quatieri, "Magnitude-only reconstruction using a sinusoidal speech model," in *Proc. ICASSP'84*, San Diego, 1984.

[MQ86] R. J. McAulay and T. F. Quatieri, "Speech analysis/synthesis based on a sinusoidal representation," *IEEE Tran. Acous. Sp. Sig. Proc.*, vol.34, no.4, 1986, pp. 744-754.

[OS89] A. V. Oppenheim and R. W. Schafer, *Discrete-time signal processing*, Prentice Hall, Englewood Cliffs NJ, 1989.

[PEF98] H. Purnhagen, B. Edler and C. Ferehdis, "Object-Based Analysis/Synthesis Audio Coder for Very Low Bit Rates," in *Proc. AES 104th Convention*, Amsterdam, 1998.

[PM00] H. Pumhagen and N. Meine, "HILN - The MPEG-4 parametric audio coding tools," in *Proc. ISCAS'00*, Geneva, 2000.

[Portnoff81] M. Portnoff, "Time-scale modification of speech based on short-time Fourier analysis," *IEEE Tran. Acous. Sp. Sig. Proc.*, vol.29, no.3, 1981.

[RD92] X. Rodet and P. Depalle, "Spectral envelope and inverse FFT synthesis," in *Proc. AES 93rd Convention*, San Francisco, 1992.

[RLV07] J. Rauhala, H.-M. Lehtonen and V. Välimäki, "Fast automatic inharmonicity estimation algorithm," *J. Acoust. Soc. Am., vol.* 121, no. 5, pp. EL184-EL189, 2007.

[Röbel02] A. Röbel, “Estimating partial frequency and frequency slope using reassignment operators,” in *Proc. ICMC’02*. Göteborg. 2002.

[Rodet97] X. Rodet, “Musical sound signal analysis/synthesis: sinusoidal+residual and elementary waveform models,” in *IEEE Time-Frequency and Time-Scale Workshop*, Coventry, 1997.

[SB90] R. L. Streit and R. F. Barrett, “Frequency line tracking using hidden Markov models,” *IEEE Tran. Acous. Sp. Sig. Proc.*, vol.38, no.4, 1990, pp. 586-598.

[Serra89] X. Serra, *A System for sound analysis/transformation/synthesis based on a deterministic plus stochastic decomposition*, PhD. Thesis, CCRMA, Stanford University, 1989.

[Serra97] X. Serra, “Musical sound modeling with sinusoids plus noise,” *Musical signal processing*, Swets & Zeitlinger Publishers, 1997.

[SW98] A. Sterian and G. H. Wakefield, “A model-based approach to partial tracking for musical transcription,” in *Proc. SPIE An. Meeting*, San Diego,1998.

[Tolonen99] T. Tolonen, “Methods for separation of harmonic sound sources using sinusoidal modeling,” in *Proc. AES 106th Convention*, Munich, 1999.

[TWRCY03] W. H. Tsai, H. M.Wang, D. Rodgers, S. S. Cheng, and H. M. Yu, “Blind clustering of popular music recordings based on singer voice characteristics,” in *Proc. ISMIR’03*, Baltimore, 2003.

[Virtanen00] T. Virtanen, *Audio Signal Modeling with Sinusoids Plus Noise*, MSc Thesis, Tampere University of Technology, 2000.

[WC92] R. Wilson, A. D. Calway and E. R. S. Pearson, “A generalized wavelet transform for Fourier analysis: the multiresolution Fourier transform and its application to image and audio signal analysis,” *IEEE Tran. Inf. Th.*, vol. 38, no.2, 1992, pp.674-690.

[WGR99] P. J. Walmsley, S. J. Godsill and P. J. W. Rayner, "Polyphonic pitch tracking using joint Bayesian estimation of multiple frame parameters," in *Proc. WASPAA'99*, New Paltz, 1999.

[WS05] Wen X. and M. Sandler, "Transcribing piano music using signal novelty," in *Proc. AES 118th Convention*, Barcelona, 2005.

[WS05b] Wen X. and M. Sandler, "A partial searching algorithm and its application for polyphonic transcription," in *Proc. ISMIR'05*, London, 2005.

[WS06] Wen X. and M. Sandler, "Error compensation in modeling time-varying sinusoids," in *Proc. DAFx'06*, Montreal, 2006.

[WS07] Wen X. and M. Sandler, "New audio editor functionality using harmonic sinusoids," in *Proc. AES122nd Convention*, Vienna, 2007.

[XE90] Xie X. and R. J. Evans, "Multiple frequency line tracking using hidden Markov models," in *Proc. 29th Conf. Decision & Control*, Honolulu, 1990.

Appendices

A. Sinusoids and Fourier transforms

B. DFT-based sinusoid estimators

C. The LSE estimator

D. Harmonic particle detector

E. Re-estimation of sinusoids

F. Spectral-domain resynthesis

Appendix A

Sinusoids and Fourier transforms

A.1 Pre-defined functions

A.1.1 Dirac Delta function

The Dirac delta function $\delta(x)$ is defined as

$$\delta(x)=\begin{cases}+\infty, & x=0\\ 0, & x\neq 0\end{cases} \text{ and } \int_{-\infty}^{+\infty}\delta(x)dx=1. \tag{A. 1}$$

A.1.2 Discrete sinc function

The N-point discrete sinc function $\text{sinc}_N(k)$ is defined as

$$\text{sinc}_N(k)=\begin{cases}(-1)^{\frac{(N-1)k}{N}}, & k=0,\pm N,\pm 2N, etc.\\ \dfrac{\sin \pi k}{N\sin\dfrac{\pi k}{N}}, & otherwise\end{cases} \tag{A. 2}$$

The discrete sinc function is the amplitude spectrum of the discrete rectangular window function, defined as

$$\mathbf{1}_{l,m}(n)=\begin{cases}1, & l\leq n<m\\ 0, & otherwise\end{cases} \tag{A. 3a}$$

Let l=0 and n=N, then the DTFT of (A.3a) is

$$\mathbf{I}_{0,N}(f)=e^{-j\pi(N-1)f}N\,\text{sinc}_N(Nf) \tag{A. 3b}$$

When N is even, $\text{sinc}_N(k)$ is a periodic function of k with period $2N$, and has zeroes at all integer k except multiples of N. There is a *main lobe* between the first pair of zeros -1 and +1. When |*k*|<<*N*, the envelope of sinc$_N$(*k*) drops from the maximum 0 like 1/*k*, and approximates the following *continuous sinc function*

$$\mathrm{sinc}(f)=\begin{cases}1, & f=0,\\ \dfrac{\sin \pi f}{\pi f}, & otherwise\end{cases} \tag{A. 4a}$$

by the relation

$$\left|\mathrm{sinc}_N(k)-\mathrm{sinc}(k)\right|<\frac{\pi^2}{2}\left(\frac{k}{N}\right)^2. \tag{A. 4b}$$

A.2 Cosine window family

A window function *w* supported on [a, b] is a member of cosine window family if

$$w(t)=\begin{cases}\sum_m a_m \cos 2\pi m\dfrac{t-(a+b)/2}{b-a}, & a\le t\le b\\ 0, & otherwise\end{cases} \tag{A. 5a}$$

This definition covers all regular functions that are symmetric on [a, b]. A narrower definition requires that *m* be limited below some small integer *M*, and *w* be low-pass.

The discrete version the cosine window family is obtained by directly sampling *w*(t). Throughout this thesis the size of a discrete window function, say *N*, is always an even number:

$$w_n=\begin{cases}\sum_{m=0}^{M-1}(-1)^m a_m \cos 2\pi mn/N, & -N/2\le n\le N/2-1\\ 0, & otherwise\end{cases} \tag{A. 5b}$$

Let $c_0= a_0$, $c_m=c_{-\mathrm{m}}=a_m/2$, then

$$w_n=\begin{cases}\sum_{m=-M+1}^{M-1}(-1)^m c_m e^{j2\pi mn/N}, & -N/2\le n\le N/2-1\\ 0, & otherwise\end{cases} \tag{A. 5c}$$

The DTFT is

$$W(f)=Ne^{-j\pi fN}\sum_{m}c_m e^{-j\pi(m/N-f)}\operatorname{sinc}_N(m-Nf)=Ne^{-j\pi fN}H(f) \qquad \text{(A.5d)}$$

or

$$W(f)=Ne^{-j\pi fN}H(f),\ H(f)=\sum_{m}c_m e^{-j\pi(m/N-f)}\operatorname{sinc}_N(m-Nf) \qquad \text{(A.5e)}$$

Some typical window functions of this family are listed in the following table.

Name	Expression on 0≤*n*<N	c_0	c_1	c_2
Rectangular	$w_n=1$	1	-	-
Hann	$w_n=0.5-0.5\cos\frac{2\pi n}{N}$	0.5	0.25	-
Hamming	$w_n=0.54-0.46\cos\frac{2\pi n}{N}$	0.54	0.23	-
Blackman	$w_n=0.42-0.5\cos\frac{2\pi n}{N}+0.08\cos\frac{4\pi n}{N}$	0.42	0.25	0.04

Table A. 1 Cosine window functions

A.3 Proofs of propositions in Chapter 1

A.3.1 Proposition 1.1

Proposition 1.1 Let *x* be a slow-varying sinusoid $x_n=a_ne^{j\varphi_n}$, then its windowed DFT can be written as

$$X_k=a_{N/2}e^{j\varphi_{N/2}}W\left(k/N-f_{N/2}\right)+\varepsilon_k, \qquad \text{(1. 24a)}$$

where the term ε_k is bounded by

$$|\varepsilon_k|\le\sup|f'|\cdot\pi a_{N/2}\sum_{n=0}^{N-1}|w_n|(n-N/2)^2+\sup|\Delta a|\cdot\sum_{n=0}^{N-1}|w_n(n-N/2)|. \qquad \text{(1. 24b)}$$

Let $\tilde{x}$ be a constant sinusoid

$$\tilde{x}_n=a_{N/2}e^{j(\varphi_{N/2}+2\pi(n-N/2)f_{N/2})}, \qquad \text{(A. 6a)}$$

and α_n, β_n be defined as

$$\alpha_n = \begin{cases} \dfrac{a_n - a_{N/2}}{n - N/2}, & n \neq N/2 \\ 0, & n = N/2 \end{cases},$$

$$\beta_n = \begin{cases} 2\dfrac{\varphi_n - \varphi_{N/2} - 2\pi f_{N/2}(n - N/2)}{(n - N/2)^2}, & n \neq N/2 \\ 0, & n = N/2 \end{cases}, \tag{A. 6b}$$

the mean-value theorem ensures that

$$|\alpha_n| \leq \sup|\Delta a|,\ \beta_n \leq 2\pi \sup|f'|. \tag{A. 6c}$$

So

$$\begin{aligned} x_n - \widetilde{x}_n &= \left(a_{N/2} + (n - N/2)\alpha_n\right)e^{j\varphi_n} - a_{N/2}e^{j(\varphi_{N/2}+2\pi(n-N/2)f_{N/2})} \\ &= a_{N/2}\left(e^{j\varphi_n} - e^{j(\varphi_{N/2}+2\pi(n-N/2)f_{N/2})}\right) + (n - N/2)\alpha_n e^{j\varphi_n} \\ &= a_{N/2}\left(e^{j(\varphi_{N/2}+2\pi f_{N/2}(n-N/2)+\beta_n(n-N/2)^2/2)} - e^{j(\varphi_{N/2}+2\pi(n-N/2)f_{N/2})}\right) + (n - N/2)\alpha_n e^{j\varphi_n} \\ &= a_{N/2}e^{j(\varphi_{N/2}+2\pi(n-N/2)f_{N/2})}\left(e^{j\beta_n(n-N/2)^2/2} - 1\right) + (n - N/2)\alpha_n e^{j\varphi_n} \end{aligned} \tag{A. 6d}$$

using the inequality x^2+2cos$x$>2, it is easy to show that

$$\left|e^{jx} - 1\right| < |x| \tag{A. 6e}$$

So (A.6d) can be written as

$$x_n - \widetilde{x}_n = a_{N/2}e^{j(\varphi_{N/2}+2\pi(n-N/2)f_{N/2})}\beta_n\gamma_n(n - N/2)^2/2 + (n - N/2)\alpha_n e^{j\varphi_n} \tag{A. 6f}$$

where $|\gamma_n| \leq 1$. Now we calculate the windowed DFT

$$\left|X_k - \tilde{X}_k\right| = \left|\sum_{n=0}^{N-1}\left(x_n - \tilde{x}_n\right)w_n e^{-j2\pi kn/N}\right|$$

$$\le a_{N/2}\sum_{n=0}^{N-1}\left|\beta_n\gamma_n w_n(n-N/2)^2/2\right| + \sum_{n=0}^{N-1}\left|(n-N/2)\alpha_n w_n\right| \qquad \text{(A.6g)}$$

$$\le a_{N/2}\pi\sup\left|f'\right|\sum_{n=0}^{N-1}\left|w_n\right|(n-N/2)^2 + \sup\left|\Delta a\right|\sum_{n=0}^{N-1}\left|w_n(n-N/2)\right|$$

This together with (1.22b) concludes the proof.■

A.3.2 Proposition 1.2

Proposition 1.2 Let x be a sinusoid with constant amplitude and varying frequency, i.e. $x_n = e^{j\varphi_n}$, its instantaneous frequency f be within the interval F=(f_1, f_2) during [0, N], and X_k be its DFT. Given an integer L, 3≤L<<N, then the amplitude spectrum at k is bounded by

$$\left|X_k\right| < \left(0.01 + 0.112/L\right)N + \frac{\pi^3 N^3}{60L^2}\sup\left|f'\right| \qquad \text{(1. 16)}$$

if k is at least 1.5M bins from F, i.e. 0≤k<Nf_1-1.5M or Nf_2+1.5M<k<N/2-1.

Let M=N/L. We divide the rectangular window $\mathbf{1}_{[0,\,N]}$ into L-1 parts as $\mathbf{1} = \sum_{l=1}^{L-1} w_l$, where each part w_l is a window function of size 2M, centred at l·M, and

$$w_{l,n} = \begin{cases} \sin^2\dfrac{\pi(n-lM+M)}{2M}, & (m-1)L < n < (m+1)L \\ 0, & otherwise \end{cases}, \; l=2,3,\ldots,L\text{-}2,$$

$$w_{1,n} = \begin{cases} 1, & 0 \le n < M \\ \sin^2\dfrac{\pi n}{2M}, & M \le n < 2M \\ 0, & otherwise \end{cases}, \quad w_{L-1,n} = \begin{cases} 1, & N-M < n < N \\ \sin^2\dfrac{\pi(n-N)}{2M}, & N-2M < n \le N-M \\ 0, & otherwise \end{cases}$$

$$\text{(A. 7a)}$$

For l=2, 3, …, L-2, w_l is a Hann window supported on [(l-1)M, (l+1)M]; for l=0 and l=L-1, w_l is a fade-out or fade-in window involving half unity and half Hann window. Let A$_l$ and B$_l$ be the left and right boundaries of w_l. Now we have

$$x_n = \sum_{l=1}^{L-1} x_n w_{l,n} \ , \ X(f) = \sum_{l=1}^{L-1} X_l(f) \tag{A. 7b}$$

where X_l is the DTFT of $x{\cdot}w_l$. According to Proposition 1.1, we have

$$X_l(f) = e^{j\varphi_{mL}} W_l\left(f - f_{lM}\right) + \varepsilon_l(f) \tag{A. 7c}$$

where

$$\left|\varepsilon_l(f)\right| \le \sup_l \left|f'\right| \cdot \pi \sum_{n=A_l}^{B_l-1} \left|w_n\right| (n - lM)^2 \tag{A. 7d}$$

$\sup_l$ is a simple form of $\sup_{A_l<n<B_l}$. It follows that

$$\begin{aligned} \left|X(f)\right| &= \left|\sum_{l=1}^{L-1} e^{j\varphi_{lM}} W_l\left(f - f_{lM}\right) + \sum_{l=1}^{L-1} \varepsilon_l(f)\right| \\ &\le \sum_{l=1}^{L-1} \left|W_l\left(f - f_{lM}\right)\right| + \pi \sum_{l=1}^{L-1} \sup_l \left|f'\right| \sum_{n=A_l}^{B_l-1} \left|w_{l,n}\right| (n - lM)^2 \end{aligned} \tag{A. 7e}$$

For the first term, when 2≤l≤L-2, w_l is a 2M-point Hann window, whose half main lobe width is L/N. So if f is more than 1.5L/N from F, then $\left|W_l\left(f - f_{lM}\right)\right|$ is below the second side-lobe peak (-41dB), i.e. $\left|W_l\left(f - f_{lM}\right)\right| < 0.01N/L$. When l=1 or l=L-1, $\left|W_l\left(f - f_{lM}\right)\right|$ has -23dB drop at 1.5L/N, i.e. $\left|W_l\left(f - f_{lM}\right)\right| < 0.71N/L$. Then

$$\sum_{l=1}^{L-1} \left|W_l\left(f - f_{lM}\right)\right| < (L-3) \cdot 0.01N/L + 2 \times 0.071N/L = (0.01 + 0.112/L)N \ . \tag{A. 7f}$$

For the second term we have

$$\begin{aligned} \sum_{n=A_l}^{B_l-1} \left|w_{l,n}\right| (n - lM)^2 &= \sum_{n=-M}^{M-1} \sin^2 \frac{\pi(n+M)}{2M} \cdot n^2 \\ &< 2 \sum_{n=-M}^{-1} \left(\frac{\pi\left(n+M\right)}{2M}\right)^2 n^2 = \frac{\pi^2 (M-1)(M+1)(M^2+1)}{60M} \end{aligned} \tag{A. 7g}$$

where we have used the equation

$$\sum_{n=1}^{N-1} n^2 (N-n)^2 = \frac{(N-1)N(N+1)(N^2+1)}{30}.$$

Finally

$$\pi \sum_{l=1}^{L-1} \sup_l |f'| \sum_{n=A_l}^{B_l-1} |w_{l,n}| (n-lM)^2$$

$$< \frac{\pi^3 (M-1)(M+1)(M^2+1)}{60M} \left(\frac{\sup_1 |f'| + \sup_{L-1} |f'|}{2} + \sum_{l=2}^{L-2} \sup_l |f'| \right) \qquad \text{(A. 7h)}$$

$$< \frac{\pi^3 M^3}{60} \left(\frac{\sup_1 |f'| + \sup_{L-1} |f'|}{2} + \sum_{l=2}^{L-2} \sup_l |f'| \right) \le \frac{\pi^3 N^3}{60L^2} \sup |f'|$$

This, together with (A.7e) and (A.7f), concludes the proof.■

Corollary 1.3 is proved by using Hann windows for *l*=1 and *l*=L-1 instead of the fade-out and fade-in windows. In this case (A.7f) is replaced by

$$\sum_{l=1}^{L-1} \left| W_l (f - f_{lM}) \right| < (L-1) \cdot 0.01N / L < 0.01N. \qquad \text{(A. 7i)}$$

The rest of the proof is the same.

A.3.3 Proposition 1.4

Proposition 1.4 Let *x*=*a*cos*φ*=*b*cos*θ*, *a*>0, *b*>0, $0 \le \frac{|a'|}{a}, \frac{|b'|}{b} \le A_1$, $0 \le \frac{|a''|}{a}, \frac{|b''|}{b} \le A_2$, $\varphi' = 2\pi f > 0$, $\theta' = 2\pi g > 0$, $0 \le |f'|, |g'| \le F_1$, $\varphi(-\tau) = \theta(-\tau) = -\frac{\pi}{2}$, $\varphi(\tau) = \theta(\tau) = \frac{\pi}{2}$,

then

$$|f-g| \le \min\left(\frac{A_2 + 2\pi(g \tan|\theta| + f \tan|\varphi|)A_1 + \pi(\tan|\theta| + \tan|\varphi|)F_1}{2\pi^2 (f+g)}, 2\tau F_1 \right), \qquad \text{(1. 29a)}$$

$$|\varphi - \theta| \le 2\pi\tau^2 F_1, \qquad \text{(1. 29b)}$$

$$\frac{|a-b|}{b} \le \frac{\sin\max(|\theta|, |\varphi|)}{\cos\varphi} 2\pi\tau^2 F_1, \text{ when } x \ne 0. \qquad \text{(1. 29c)}$$

To prove (1.29a), we calculate the derivatives of $a\cos\varphi=b\cos\theta$:

$$a'\cos\varphi - 2\pi af\sin\varphi = b'\cos\theta - 2\pi bg\sin\theta \tag{A. 8a}$$

$$\begin{aligned} &a''\cos\varphi - 4\pi a'f\sin\varphi - 2\pi af'\sin\varphi - 4\pi^2 xf^2 \\ &= b''\cos\theta - 4\pi b'g\sin\theta - 2\pi bg'\sin\theta - 4\pi^2 xg^2 \end{aligned} \tag{A. 8b}$$

then

$$\begin{aligned} &\left|f^2 - g^2\right| \\ &= \frac{\left|a''\cos\varphi - 4\pi a'f\sin\varphi - 2\pi af'\sin\varphi - b''\cos\theta + 4\pi b'g\sin\theta + 2\pi bg'\sin\theta\right|}{4\pi^2 x} \\ &\le \left|\frac{1}{4\pi^2}\frac{a''}{a} - \frac{1}{4\pi^2}\frac{b''}{b} + \frac{g\tan\theta}{\pi}\frac{b'}{b} - \frac{f\tan\varphi}{\pi}\frac{a'}{a} + \frac{\tan\theta}{2\pi}g' - \frac{\tan\varphi}{2\pi}f'\right| \\ &\le \frac{1}{2\pi^2}A_2 + \frac{g\tan|\theta| + f\tan|\varphi|}{\pi}A_1 + \frac{\tan|\theta| + \tan|\varphi|}{2\pi}F_1 \end{aligned} \tag{A. 8c}$$

so

$$|f-g| \le \frac{A_2 + 2\pi(g\tan|\theta| + f\tan|\varphi|)A_1 + \pi(\tan|\theta| + \tan|\varphi|)F_1}{2\pi^2(f+g)} \tag{A. 8d}$$

To show the other half, let $h=f-g$, apparently $|h'| \le 2F_1$, and

$$\int_{-\tau}^{\tau} h(t)dt = 0 . \tag{A. 8e}$$

$\forall\, t_0 \in (-\tau, \tau)$,

$$\begin{aligned} &\left|-2\tau\cdot h(t_0)\right| = \left|\int_{-\tau}^{\tau}\left(h(t) - h(t_0)\right)dt\right| \\ &\le \sup\left|h'\right|\int_{-\tau}^{\tau}\left|t - t_0\right|dt = \sup\left|h'\right|\left(t_0^2 + \tau^2\right) \le 2F_1\left(t_0^2 + \tau^2\right) \end{aligned} \tag{A. 8f}$$

$$\left|h(t_0)\right| \le \frac{2F_1\left(t_0^2 + \tau^2\right)}{2\tau} \le 2\tau F_1 . \tag{A. 8g}$$

This, together with (A.8d), conclude the proof of (1.29a).

To show (1.29b), let $\psi = \varphi - \theta$, then $\psi(-\tau) = \psi(\tau) = 0$, $\psi' = 2\pi h$. Obviously $|\psi|$ has its maxima at a zero of h. Let it be t_0, then

$$| \psi(t_0) | = \left| 2\pi \int_{-\tau}^{t_0} h(t)dt \right| \le 2\pi \int_{-\tau}^{t_0} \sup|h'||t - t_0|dt \le 2\pi(t_0{+}\tau)^2 F_1 ,$$

$$| \psi(t_0) | = \left| 2\pi \int_{\tau}^{t_0} h(t)dt \right| \le 2\pi(t_0{-}\tau)^2 F_1 \tag{A. 8h}$$

Therefore

$$| \varphi - \theta | \le | \psi(t_0) | \le 2\pi F_1 \min\left((t_0{+}\tau)^2, (t_0{-}\tau)^2\right) \le 2\pi\tau^2 F_1 \tag{A. 8i}$$

From (A.8i) we directly have

$$\frac{|a-b|}{b} = \frac{|\cos\theta - \cos\varphi|}{\cos\varphi} \le \frac{\sin\max(|\theta|,|\varphi|)}{\cos\varphi}|\theta - \varphi| \le \frac{\sin\max(|\theta|,|\varphi|)}{\cos\varphi} 2\pi\tau^2 F_1 \tag{A. 8j}$$

since sin function is monotonic on [–π/2, π/2]. ■

A.3.4 Slowest-varying sinusoids

Proof of 1.30c: The necessary condition to minimize

$$I = \int (\eta a'(t)^2 + (1-\eta)\varphi''(t)^2) dt \tag{1. 30a}$$

subject to the condition

$$a(t)\cos\varphi(t) = x(t) . \tag{A. 9a}$$

We use the Lagrange multiplier λ(t). Let

$$F(t, a, a', \varphi, \varphi', \varphi'') = \eta\left(a'\right)^2 + (1-\eta)\left(\varphi''\right)^2 ,\ G(t, a, \varphi) = a\cos\varphi - x , \tag{A. 9b}$$

and

$$L(t, a, a', \varphi, \varphi'') = F + \lambda G = \eta(a')^2 + (1-\eta)(\varphi'')^2 - \lambda(a\cos\varphi - x) , \tag{A. 9c}$$

$L(\cdot)$ involves t, a, a', φ, φ'', the Euler equation for extrema of integral (1.30a) is

$$L_a - \frac{d}{dt}L_{a'} = L_\varphi + \frac{d^2}{dt^2}L_{\varphi''} = 0 . \tag{A. 9d}$$

From (A.9d) we have

$$L_a = -\lambda \cos\varphi\,, \tag{A. 9e}$$

$$L_{a'} = 2\eta a'\,, (d/dt)L_{a'} = 2\eta a''\,, \tag{A. 9f}$$

$$L_\varphi = \lambda a \sin\varphi\,, \tag{A. 9g}$$

$$L_{\varphi''} = 2\varphi''\,, (d^2/dt^2)L_{\varphi''} = 2\varphi^{(4)} \tag{A. 9h}$$

Eliminating λ we get

$$-\eta a'' a \tan\varphi + (1-\eta)\varphi^{(4)} = 0\,. \tag{1. 30c}$$

■

Proof of 1.31b: The necessary condition to minimize

$$I = \sum_n \eta(\Delta a)^2 + (1-\eta)(\Delta^2\varphi)^2 \tag{1. 31a}$$

subject to the condition

$$a_n \cos\varphi_n = x_n\,. \tag{A. 10a}$$

Again using the Lagrange multiplier we write

$$L = \sum_n \eta(\Delta a)^2 + (1-\eta)(\Delta^2\varphi)^2 - \lambda_n(a_n \cos\varphi_n - x_n) \tag{A. 10b}$$

To minimize L we calculate the partial derivatives:

$$\frac{\partial L}{\partial a_n} = 2\eta(2a_n - a_{n-1} - a_{n+1}) - \lambda_n \cos\varphi_n = 0\,, \tag{A. 10c}$$

$$\frac{\partial L}{\partial \varphi_n} = 2(1-\eta)(6\varphi_n - 4\varphi_{n-1} - 4\varphi_{n+1} + \varphi_{n-2} + \varphi_{n+2}) + \lambda_n a_n \sin\varphi_n = 0\,. \tag{A. 10d}$$

Eliminating λ_n we get

$$\eta a_n(2a_n - a_{n-1} - a_{n+1})\tan\varphi_n + (1-\eta)(6\varphi_n - 4\varphi_{n-1} - 4\varphi_{n+1} + \varphi_{n-2} + \varphi_{n+2}) = 0\,. \tag{A. 10e}$$

By writing (A.10e) using difference operators we get (1.31b).■

Appendix B

Sinusoid measurements I:

DFT-based sinusoid estimators

B.1 Detecting sinusoids as spectral peaks

B.1.1 Proof of Proposition 2.1

Proposition 2.1 (windowed DFT): If the window spectrum $W(f)$ satisfies that $|W(f)|>|W(g)|$, $\forall\ |f|\leq 0.5/N$, $0.5/N<|g|<1\text{-}0.5/N$, $N\cdot(f-g)\in\mathrm{Z}$, then a constant complex sinusoid $x_n=ae^{j(2\pi fn+\varphi)}$ ($0\leq f<0.5$) has a global windowed DFT peak at bin k, where k/N is closest to f.

Recall equation (1.23):

$$X_k = ae^{j\varphi}\sum_n w_n e^{j2\pi(f-k/N)n} = e^{j\varphi}\cdot aW(k/N-f) \tag{1.23}$$

Let k be the integer closest to Nf, i.e. $|k/N\text{-}f\,|\leq 0.5/N$, and let l be an integer between 0 and N-1 so that $l\neq k$. Obviously $N\cdot\big((l/N-f)-(k/N-f)\big)=l-k\in\mathrm{Z}$, $0.5/N\leq|l/N\text{-}f\,|<1\text{-}0.5/N$. If $|l/N\text{-}f\,|>0.5/N$, we immediately have $|X_k|>|X_l|$, following the assumptions on W. If $|l/N\text{-}f\,|=0.5/N$, we immediately have $l/N\text{-}f=\text{-}(k/N\text{-}f\,)$. Since $|W|$ is symmetric, $|X_k|=|X_l|$. The identity holds only once for all $0\leq l<N$, $l\neq k$, which guarantees that $|X_k|=|X_{k+1}|$ and $|X_k|=|X_{k\text{-}1}|$ do not hold at the same time. Therefore bin k is a global peak.■

B.1.2 Proof of Proposition 2.2

Proposition 2.2 (noise tolerance): Let x be a complex sinusoid mixed with noise r, i.e. $x_n=ae^{j(2\pi fn+\varphi)}+r_n$ ($0\leq f<0.5$), $W(f)$ be the window spectrum, K be a positive integer, then the windowed spectrum X_k has a local peak within K bins from Nf, provided that $|R_k|<0.5aW(0)\Delta$ for Nf-1-K<k<Nf+1+K, where Δ is defined as

$$\Delta = \max_{L\in Z^+, L\le K} \Delta_L\,,\quad \Delta_L = \inf_{|f|\le 0.5/N} \frac{|W(f)| - |W(f+L/N)|}{W(0)}\,. \tag{2.2}$$

Let k be the integer closest to Nf, i.e. $|k\text{-}Nf|\le 0.5$. To show the existence of a local peak, we only need to find L, $1\le L\le K$, so that $|X_k|>|X_{k-L}|$, $|X_k|>|X_{k+L}|$. Due to the linearity of DFT, we have

$$X_k = e^{j\varphi}\cdot aW(k/N-f)+R_k\,. \tag{B. 1a}$$

Therefore

$$\begin{aligned} |X_k|-|X_{k+L}| &= \left|e^{j\varphi}\cdot aW\big(k/N-f\big)+R_k\right|-\left|e^{j\varphi}\cdot aW(k/N-f+L/N)+R_{k+L}\right| \\ &\ge a\big(|W(k/N-f)|-|W(k/N-f+L/N)|\big)-|R_k|-|R_{k+L}| \end{aligned} \tag{B. 1b}$$

and

$$\begin{aligned} |X_k|-|X_{k-L}| &\ge a\big(|W(k/N-f)|-|W(k/N-f-L/N)|\big)-|R_k|-|R_{k-L}| \\ &= a\big(|W(f-k/N)|-|W(f-k/N+L/N)|\big)-|R_k|-|R_{k-L}| \end{aligned} \tag{B. 1c}$$

where we have used the symmetry of w. Let

$$L_1 = \arg\max_{L\in Z^+, L\le K} \Delta_L\,, \tag{B. 1d}$$

i.e. $\Delta=\Delta_{L_1}$, then $L_1\in Z^+$, $L_1\le K$, and

$$\frac{|W(f)|-|W(f+L_1/N)|}{W(0)} \ge \Delta\,,\ \forall\ |f|\le 0.5/N. \tag{B. 1e}$$

Following (B.1b) and (B.1c)

$$|X_k|-|X_{k+L_1}| > aW(0)\Delta - 2\cdot 0.5aW(0)\Delta \ge 0 \tag{B. 1f}$$

$$|X_k|-|X_{k-L_1}| > aW(0)\Delta - 2\cdot 0.5aW(0)\Delta \ge 0 \tag{B. 1g}$$

These conclude the proof.■

B.1.3 Proof of Corollary 2.4

Corollary 2.4 (sinusoidal noise): Let the noise r be a constant sinusoid, i.e. $r_n=be^{j(2\pi gn+\theta)}$, then $x_n=ae^{j(2\pi fn+\varphi)}+r_n$ has a local windowed spectral peak within K bins from Nf, provided that $b<a\Delta_s$, where Δ_s is defined as

$$\Delta_s=\max_{L\in Z^+,L\leq K}\Delta_s(L),\ \Delta_s(L)=\inf_{|f|\leq 0.5/N}\left(\frac{|W(f)|-|W(f+L/N)|}{|W(f\pm h)|+|W(f\pm h+L/N)|}\right),\ h=g\text{-}f. \quad (2.\ 4)$$

Let k be the integer closest to Nf, i.e. $|k\text{-}Nf|\leq 0.5$. To show the existence of a local peak, we only need to find L, $1\leq L\leq K$, so that $|X_k|>|X_{k-L}|$, $|X_k|>|X_{k+L}|$.Calculating the DFT we get

$$X_k=ae^{j\varphi}W(k/N-f)+be^{j\theta}W(k/N-g) \quad \text{(B. 2a)}$$

Again we compare the main peak X_k with a bin L bins away:

$$\begin{aligned}|X_k|-|X_{k+L}|&=\left|ae^{j\varphi}W(k/N-f)+be^{j\theta}W(k/N-g)\right|\\&\quad-\left|ae^{j\varphi}W(k/N-f+L/N)+be^{j\theta}W(k/N-g+L/N)\right|\\&\geq\left|ae^{j\varphi}W(k/N-f)\right|-\left|be^{j\theta}W(k/N-g)\right|\\&\quad-\left|ae^{j\varphi}W(k/N-f+L/N)\right|-\left|be^{j\theta}W(k/N-g+L/N)\right|\\&=\left(a\frac{|W(k/N-f)|-|W(k/N-f+L/N)|}{|W(k/N-f-h)|+|W(k/N-f-h+L/N)|}-b\right)\cdot\\&\quad\left(|W(k/N-g)|+|W(k/N-g+L/N)|\right)\end{aligned} \quad \text{(B. 2b)}$$

$$|X_k|-|X_{k-L}|\geq\left(a\frac{|W(k/N-f)|-|W(k/N-f-L/N)|}{|W(k/N-f-h)|+|W(k/N-f-h-L/N)|}-b\right)\cdot$$
$$\left(|W(k/N-g)|+|W(k/N-g+L/N)|\right)$$
$$=\left(a\frac{|W(f-k/N)|-|W(f-k/N+L/N)|}{|W(f-k/N+h)|+|W(f-k/N+h+L/N)|}-b\right)\cdot$$
$$\left(|W(k/N-g)|+|W(k/N-g+L/N)|\right) \qquad \text{(B. 2c)}$$

where we have used the symmetry of *w*. Let

$$L_1=\arg\max_{L\in Z^+,L\leq K}\Delta_s(L) \qquad \text{(B. 2d)}$$

i.e. Δ_s=$\Delta_s(L_1)$, then $L_1 \in Z^+$, $L_1 \leq K$, and

$$\frac{|W(f)|-|W(f+L_1/N)|}{|W(f\pm h)|+|W(f\pm h+L_1/N)|}\geq\Delta_s,\ \forall\ |f|\leq 0.5/N. \qquad \text{(B. 2e)}$$

Following (B.2b) and (B.2c), we have

$$|X_k|-|X_{k+L_1}|\geq(a\Delta_s-b)\cdot\left(|W(k/N-g)|+|W(k/N-g+L/N)|\right)>0 \qquad \text{(B. 2f)}$$

$$|X_k|-|X_{k-L_1}|\geq(a\Delta_s-b)\cdot\left(|W(k/N-g)|+|W(k/N-g+L/N)|\right)>0 \qquad \text{(B. 2g)}$$

■

B.2 Frequency estimation methods

B.2.1 Proof of (2.6b)

Since the standard FFT method only concerns the power (amplitude) spectrum, we use the 0-centred DFT instead of 0-based DFT, so that *W*(*f*) and ε are both real. In the context of zero-padding $\hat{f}=k/N$ becomes a continuous variable. Combining (2.5a) and (2.5b) we get

$$X_k=\frac{a}{2}e^{j\varphi}W(\delta)+\frac{a}{2}e^{-j\varphi}W(\delta+2f) \qquad \text{(B. 3a)}$$

We calculate the power spectrum

$$|X_k|^2 = 0.25a^2\left(W^2(\delta) + W^2(\delta+2f) + 2\cos 2\varphi W(\delta)W(\delta+2f)\right) \tag{B. 3b}$$

To find the spectral peak we let $\frac{d|X_k|^2}{d\hat{f}}$=0 and get

$$W'(\delta) = -W'(\delta+2f)\frac{\varepsilon+\cos 2\varphi}{1+\varepsilon\cos 2\varphi} \tag{B. 3c}$$

It follows that

$$|W'(\delta)| \le |W'(\delta+2f)| \cdot \frac{1+|\varepsilon|}{1-|\varepsilon|} \tag{B. 3d}$$

On the other hand,

$$|W'(\delta)| = \left|\left(\sum_n w_n \cos 2\pi\delta n\right)'\right| = \left|2\pi\sum_n nw_n \sin 2\pi\delta n\right| \tag{B. 3e}$$

Under the condition $|\delta|$ <0.5/*N*, |2$\pi\delta$ *n*|≤π/2, ∀–*N*/2<*n*<*N*/2, so that in (B.3e) the term $nw_n 2\pi\delta n$ does not change sign, and

$$\frac{\sin 2\pi\delta n}{2\pi\delta n} > \frac{2}{\pi} \tag{B. 3f}$$

So that

$$\begin{aligned} |W'(\delta)| &= \left|2\pi\sum_n nw_n \cdot 2\pi\delta n\frac{\sin 2\pi\delta n}{2\pi\delta n}\right| = 4\pi^2\left(\sum_n n^2 w_n \frac{\sin 2\pi\delta n}{2\pi\delta n}\right)|\delta| \\ &> 4\pi^2\left(\sum_n n^2 w_n \frac{2}{\pi}\right)|\delta| = \left(8\pi\sum_n n^2 w_n\right)\cdot|\delta| \end{aligned} \tag{B. 3g}$$

Combining (B.3d) and (B.3g) we get

$$|\delta| = \left|\hat{f} - f\right| < \frac{|W'(\delta+2f)|}{8\pi\sum_n n^2 w_n} \cdot \frac{1+|\varepsilon|}{1-|\varepsilon|} \tag{2. 6b}$$

■

B.2.2 Spectral peak of time-varying sinusoids

$$x(n) = a(n)\exp j(\varphi_0 + 2\pi\int_0^n f(t)dt) \qquad \text{(B. 4a)}$$

We calculate its windowed DTFT

$$X(g) = \sum_{n=0}^{N-1} a_n w_n e^{j\varphi_n} e^{-j2\pi gn} \qquad \text{(B. 4b)}$$

and the power spectrum

$$|X(g)|^2 = X(g)X^*(g) = \sum_{n=0}^{N-1}\sum_{m=0}^{N-1} a_m a_n w_m w_n \cos\Delta\varphi_{mn} \qquad \text{(B. 4c)}$$

where

$$\Delta\varphi_{mn}=\varphi_n-\varphi_m-2\pi g(n-m). \qquad \text{(B. 4d)}$$

To locate the DTFT peak we let the derivative of $|X(g)|^2$ be zero at $\hat{f}$:

$$\frac{d\left|X(\hat{f})\right|^2}{d\hat{f}} = 2\pi\sum_{n=0}^{N-1}\sum_{m=0}^{N-1} w_{mn} a_m a_n \sin\Delta\varphi_{mn} = 0 \qquad \text{(B. 4e)}$$

where $w_{mn}=(n-m)w_m w_n$. Using the equation $\sin\pi x = \pi x \cdot \operatorname{sinc} x$, we get

$$\hat{f} = \frac{\sum_{n=0}^{N-1}\sum_{m=0}^{N-1} w_{mn} a_m a_n \operatorname{sinc}\frac{\Delta\varphi_{mn}}{\pi}\cdot\int_m^n f(t)dt}{\sum_{n=0}^{N-1}\sum_{m=0}^{N-1} w_{mn} a_m a_n \operatorname{sinc}\frac{\Delta\varphi_{mn}}{\pi}\cdot(n-m)} . \qquad \text{(2. 7a)}$$

After some rearrangement of the summing indices, we get

$$\hat{f} = \frac{\sum_{l=0}^{N-2}\eta_l\int_0^1 f(l+t)dt}{\sum_{l=0}^{N-2}\eta_l}, \quad \eta_l = \sum_{n=l+1}^{N-1}\sum_{m=0}^{l} w_{mn} a_m a_n \operatorname{sinc}\frac{\Delta\varphi_{mn}}{\pi} \qquad \text{(2. 7b)}$$

■

B.2.3 Frequency reassignment method

The Frequency reassignment method estimate the frequency as

$$\hat{f} = \frac{k}{N} - \frac{1}{2\pi} \mathrm{Im} \frac{X_k^{w'}}{X_k^{w}} \tag{2. 11}$$

Let w_n, w_n' be the sampled versions of $w(t)$, $w'(t)$, with sampling period 1, and $W(f)$, $W^{(1)}(f)$ be their DTFTs. We have

$$W_c'(f) = j2\pi f \cdot W_c(f) \tag{B. 5a}$$

$$W(f) = \sum_{n=-\infty}^{\infty} W_c(f-n) \tag{B. 5b}$$

$$W^{(1)}(f) = \sum_{n=-\infty}^{\infty} W_c'(f-n) = \sum_{n=-\infty}^{\infty} j2\pi(f-n) \cdot W_c(f-n) \tag{B. 5c}$$

These finally yield

$$\frac{W^{(1)}(f)}{W(f)} = j2\pi \frac{\sum_{n=-\infty}^{\infty} (f-n) \cdot W_c(f-n)}{\sum_{n=-\infty}^{\infty} W_c(f-n)} \tag{B. 5d}$$

As we consider real symmetric windows only, $W_c(f)$ is real. So

$$\mathrm{Im} \frac{W^{(1)}(f)}{W(f)} = 2\pi \frac{\sum_{n=-\infty}^{\infty} (f-n) \cdot W_c(f-n)}{\sum_{n=-\infty}^{\infty} W_c(f-n)} \tag{B. 5e}$$

Given equation (1.23)[1] we have

$$X_k^w = ae^{j\varphi} W(k/N - f) \text{ and } X_k^{w'} = ae^{j\varphi} W^{(1)}(k/N - f) \tag{B. 5f}$$

Therefore

$$\hat{f} = \frac{k}{N} - \frac{\sum_{n=-\infty}^{\infty} (k/N - f - n) \cdot W_c(k/N - f - n)}{\sum_{n=-\infty}^{\infty} W_c(k/N - f - n)} = f + \frac{\sum_{n=-\infty}^{\infty} n \cdot W_c(\delta - n)}{\sum_{n=-\infty}^{\infty} W_c(\delta - n)} \tag{B. 5g}$$

where $\delta = k/N - f$. The frequency estimation error is

[1] A linear phase shift (which corresponds to a half-frame time shift) is omitted here. This does not affect the outcome of (B.5g).

$$\hat{f}-f=\frac{\sum_{n=-\infty}^{\infty} n\cdot W_c(\delta-n)}{\sum_{n=-\infty}^{\infty} W_c(\delta-n)}. \tag{2. 12}$$

■

B.2.4 Derivative method

Let x be a constant complex sinusoid, then

$$\Delta x_n = x_n - x_{n-1} = x_n(1-e^{-j2\pi f}) \tag{B. 6a}$$

$$X_k^0 = ae^{j\varphi}W(k/N-f),\ X_k^1 = ae^{j\varphi}W(k/N-f)(1-e^{-j2\pi f}) \tag{B. 6b}$$

so that

$$\frac{X_k^1}{X_k^0}=1-e^{-j2\pi f}=e^{-j\pi f}\cdot j2\sin\pi f \tag{B. 6c}$$

This leads to (2.15), which estimates the frequency without any error. Now let x be a real sinusoid, so that

$$\frac{X_k^0}{a}=e^{j\varphi}W(\delta)+e^{-j\varphi}W(\delta+2f),$$

$$\frac{X_k^1}{a}=(1-e^{-j2\pi f})e^{j\varphi}W(\delta)+(1-e^{j2\pi f})e^{-j\varphi}W(\delta+2f) \tag{B. 7b}$$

where we have substituted δ=k/N-f, and φ is taken at the window centre. Then

$$\frac{X_k^1}{X_k^0}=\left(1-e^{-j2\pi f}\right)\cdot\frac{1-e^{-j2\varphi}e^{j2\pi f}\varepsilon}{1+e^{-j2\varphi}\varepsilon}=e^{-j\pi f}\cdot j2\sin\pi f\cdot\left(1-\varepsilon\frac{e^{-j2\varphi}(1+e^{j2\pi f})}{1+e^{-j2\varphi}\varepsilon}\right) \tag{B. 7c}$$

where $\varepsilon=\dfrac{W(\delta+2f)}{W(\delta)}$. Accordingly

$$\left|\frac{1}{2}\left|\frac{X_k^1}{X_k^0}\right|-\left|\sin\pi f\right|\right|\le\left|\sin\pi f\right|\frac{2|\varepsilon|}{1-|\varepsilon|} \tag{B. 7d}$$

when f is not too low, i.e. $|\varepsilon|$<<1. The estimation error is then

$$\left|\hat{f}-f\right| = \frac{1}{\pi}\left|\arcsin\left(\frac{1}{2}\left|\frac{X_k^1}{X_k^0}\right|\right) - \arcsin\left(\left|\sin \pi f\right|\right)\right|$$

$$= \frac{1}{\pi\sqrt{1-\left(\left|\sin\pi f\right| + \theta\left(\frac{1}{2}\left|\frac{X_k^1}{X_k^0}\right| - \left|\sin\pi f\right|\right)\right)^2}}\left|\frac{1}{2}\left|\frac{X_k^1}{X_k^0}\right| - \left|\sin\pi f\right|\right|$$

$$\le \frac{1}{\pi}\frac{\left|\sin\pi f\right|\frac{2|\varepsilon|}{1-|\varepsilon|}}{\sqrt{1-\left(\left|\sin\pi f\right|\frac{1+|\varepsilon|}{1-|\varepsilon|}\right)^2}} = \frac{2\varepsilon}{\pi}\sqrt{\frac{\left|\sin\pi f\right|^2}{(1-\varepsilon)^2 - \left|\sin\pi f\right|^2(1+\varepsilon)^2}} \le \frac{2\left|\tan\pi f\right|}{\pi}|\varepsilon| \tag{B. 7e}$$

where 0≤θ≤1 is number introduced from using the mean value theorem. This concludes the proof of (2.16). ■

B.2.4 Phase vocoder (phase difference) method

Now let x be a constant real sinusoid, i.e. a complex sinusoid plus its conjugate, then

$$X_k = a\left(e^{j\varphi}W(\delta) + e^{-j\varphi}W(\delta+2f)\right),$$

$$X_k^1 = a\left(e^{j(\varphi+2\pi f)}W(\delta) + e^{-j(\varphi+2\pi f)}W(\delta+2f)\right) \tag{B. 8a}$$

It follows that

$$\begin{aligned}\arg\frac{X_k^1}{X_k} &= \arg\frac{e^{j(\varphi+2\pi f)}W(\delta) + e^{-j(\varphi+2\pi f)}W(\delta+2f)}{e^{j\varphi}W(\delta) + e^{-j\varphi}W(\delta+2f)} \\ &= \arg e^{j2\pi f}\left(1-\varepsilon\frac{e^{-j2\varphi}\left(1-e^{-j4\pi f}\right)}{1+e^{-j2\varphi}\varepsilon}\right) = \arg e^{j2\pi f}\left(\frac{1+e^{-j2\varphi}e^{-j4\pi f}\varepsilon}{1+e^{-j2\varphi}\varepsilon}\right)\end{aligned} \tag{B. 8b}$$

When f is well above the pass-band of W, i.e. $|\varepsilon|$<<1, the frequency estimation error is

$$\left|\hat{f}-f\right| = \frac{1}{2\pi}\left|\arg\frac{X_k^1}{X_k} - \arg e^{j2\pi f}\right| = \left|\arg\frac{1+e^{-j2\varphi}e^{-j4\pi f}\varepsilon}{1+e^{-j2\varphi}\varepsilon}\right| \le 2\arcsin|\varepsilon| \tag{B. 8c}$$

where the inequality is shown by notice the numerator and denominator being points on a circle centred at 1 with radius $|\varepsilon|$. This concludes the proof of (2.18). ■

Appendix C

Sinusoid measurements II:

The LSE sinusoid estimator

C.1 Computation of the sinc function and its derivatives

To find the optimal frequency of a sinusoid in the LSE sense, one needs to calculate the sinc function defined as a periodical function with period 2*N* and

$$\mathrm{sinc}_N(x) = \begin{cases} 1, \quad x = 0 \\ (-1)^{N-1}, \quad x = N \\ \dfrac{\sin \pi x}{N \sin(\pi x / N)}, \quad -N < x < N,\, x \neq 0 \end{cases} \qquad , \qquad \text{(C. 1)}$$

as well as its 1$^{\text{st}}$- and 2$^{\text{nd}}$-order derivatives. 1$^{\text{st}}$-order derivative is necessary for using gradient method of optimization, and 2$^{\text{nd}}$-order derivative is necessary for Newton or conjugate gradient methods. We calculate only for $|x|$<<*N*.

There is a numerical risk in calculating the sinc function when *x* is very close to zero. Since sin($\pi x/N$) is *N* times smaller than the numerator, it underflows before the numerator does. So instead of comparing *x* with 0, We compute sin($\pi x/N$) first then compare it with 0. The computation of the sinc function is as follows.

$$\boxed{\begin{array}{l} \text{If } \sin(\pi x / N) = 0, \mathrm{sinc}_N(x) = 1; \\[2ex] \text{else } \ s\mathrm{inc}_N(x) = \dfrac{\sin \pi x}{N \sin(\pi x / N)}. \end{array}}$$

The 1st-order derivative of the sinc function is periodical with 2N and

$$\operatorname{sinc}_N'(x) = \begin{cases} 0, \quad x = 0 \text{ or } x = N \\ \dfrac{\pi}{N} \dfrac{\sin\dfrac{\pi x}{N}\cos\pi x - \dfrac{1}{N}\sin\pi x\cos\dfrac{\pi x}{N}}{\sin^2\dfrac{\pi x}{N}}, \quad -N < x < N, x \neq 0 \end{cases} \tag{C. 2a}$$

There is a new numerical risk in the subtraction. When x is close to zero, $\sin\frac{\pi x}{N}\cos\pi x$ is the magnitude of $\pi x/N$, yet the difference in the numerator is the magnitude of $(\pi x)^3/3N$. Therefore the subtraction result is likely to suffer considerable loss of precision. Dividing this result by $\sin^2\frac{\pi x}{N}$ may incur large error. We expand $f(\pi x) = \sin\frac{\pi x}{N}\cos\pi x - \frac{1}{N}\sin\pi x\cos\frac{\pi x}{N}$ in the Taylor form in the vicinity of 0. To do this we calculate

$$f'(\pi x) = \frac{1}{2}\left(1-\frac{1}{N^2}\right)\left(\cos\left(1+\frac{1}{N}\right)\pi x - \cos\left(1-\frac{1}{N}\right)\pi x\right), \tag{C. 2b}$$

$$f''(\pi x) = \frac{1}{2}\left(1-\frac{1}{N^2}\right)\left(-\left(1+\frac{1}{N}\right)\sin\left(1+\frac{1}{N}\right)\pi x + \left(1-\frac{1}{N}\right)\sin\left(1-\frac{1}{N}\right)\pi x\right), \tag{C. 2c}$$

$$f'''(\pi x) = \frac{1}{2}\left(1-\frac{1}{N^2}\right)\left(-\left(1+\frac{1}{N}\right)^2\cos\left(1+\frac{1}{N}\right)\pi x + \left(1-\frac{1}{N}\right)^2\cos\left(1-\frac{1}{N}\right)\pi x\right), \tag{C. 2d}$$

$$f^{(4)}(\pi x) = \frac{1}{2}\left(1-\frac{1}{N^2}\right)\left(\left(1+\frac{1}{N}\right)^3\sin\left(1+\frac{1}{N}\right)\pi x - \left(1-\frac{1}{N}\right)^3\sin\left(1-\frac{1}{N}\right)\pi x\right). \tag{C. 2e}$$

Hence

$$f'(0) = 0,\ f''(0) = 0,\ f'''(0) = -\frac{2}{N}\left(1-\frac{1}{N^2}\right) \tag{C. 2f}$$

So the Taylor form is

$$\sin\frac{\pi x}{N}\cos\pi x - \frac{1}{N}\sin\pi x\cos\frac{\pi x}{N} = -\frac{(\pi x)^3}{3N}\left(1-\frac{1}{N^2}\right) + \frac{f^{(4)}(\theta\pi x)}{24}(\pi x)^4, \quad 0 \le \theta \le 1 \tag{C. 2g}$$

When |x| becomes smaller than some threshold, we use $-\frac{(\pi x)^3}{3N}\left(1-\frac{1}{N^2}\right)$ instead of $\sin\frac{\pi x}{N}\cos\pi x-\frac{1}{N}\sin\pi x\cos\frac{\pi x}{N}$ for the calculation. The switching point is determined by comparing the error bounds of the two computations. The error of direct computation comes from limited word length. Let it be L. In the floating point case both the minuend and the subtrahend have an error bound 2^{-L-M}, where $0.5\le 2^M\pi x/N<1$. The subtraction result therefore has an error bound $4{\cdot}2^{-L}{\cdot}\frac{\pi x}{N}$. The error of the trinomial computation comes from the residue, which can be estimated as

$$\frac{f^{(4)}(\theta\pi x)}{24}(\pi x)^4=\frac{1}{48}\frac{N^2-1}{N^2}\left(\left(\frac{N+1}{N}\right)^3\sin\frac{N+1}{N}\theta\pi x-\left(\frac{N-1}{N}\right)^3\sin\frac{N-1}{N}\theta\pi x\right)(\pi x)^4$$

$$\cong\frac{1}{6N}\left(1-\frac{1}{N^4}\right)\theta(\pi x)^5 \tag{C. 2h}$$

which has a bound $\frac{(\pi x)^4}{6}\frac{\pi x}{N}$. The trinomial function has a lower error bound when $(\pi x)^4<24{\cdot}2^{-L}$. The switch point can therefore be chosen at $x=\pi^{-1}(24\cdot 2^{-L})^{1/4}$. For Intel 64-bit double precision floating point, L=53, then the switching point can be chosen at 7.23×10^{-5}.

However, the switching between the two computation methods leads to discontinuity at the switching point. To preserve continuity, we use a switching interval with overlap-add. That is, we pick a pair of thresholds Th_1, Th_2, $0<Th_1<Th_2$, and preferably they lie on different sides of the ideal switching point. When $|x|>Th_2$ we calculate $\sin\frac{\pi x}{N}\cos\pi x-\frac{1}{N}\sin\pi x\cos\frac{\pi x}{N}$ directly; when $|x|<Th_1$ we calculate it using the trinomial; in between the two thresholds we linearly interpolate the two. The routine for computing $f(\pi x)$ is

$$\text{if } |x| < Th_1, f(\pi x) = -\frac{(\pi x)^3}{3N}\left(1-\frac{1}{N^2}\right)$$

$$\text{else if } |x| < Th_2,$$

$$f(\pi x) = \frac{|x|-Th_1}{Th_2-Th_1}\left(\sin\frac{\pi x}{N}\cos\pi x - \frac{1}{N}\sin\pi x\cos\frac{\pi x}{N}\right) - \frac{Th_2-|x|}{Th_2-Th_1}\frac{(\pi x)^3}{3N}\left(1-\frac{1}{N^2}\right)$$

$$\text{else } f(\pi x) = \sin\frac{\pi x}{N}\cos\pi x - \frac{1}{N}\sin\pi x\cos\frac{\pi x}{N}$$

The complete routine for calculating the 1st-order derivative is

$$\text{if } \sin(\pi x/N) = 0, \text{sinc}_N{}'(x) = 0;$$

$$\text{else calculate } f(\pi x), \text{ then } \text{sinc}_N{}'(x) = \frac{\pi}{N\sin^2(\pi x/N)} f(\pi x)$$

The 2nd-order derivative of the sinc function is periodical with 2N and

$$\text{sinc}_N{}''(x) = \begin{cases} -\dfrac{\pi^2}{3}\left(1-\dfrac{1}{N^2}\right), & x=0 \\[2ex] \dfrac{\pi^2}{3}\left(1-\dfrac{1}{N^2}\right)(-1)^N, & x=N \\[2ex] \dfrac{\pi^2}{N\sin^3(\pi x/N)}\left(\left(-1+\dfrac{1}{N^2}\right)\sin^2\dfrac{\pi x}{N}\sin\pi x - \dfrac{2}{N}\cos\dfrac{\pi x}{N}\cdot\right. \\[2ex] \qquad \left.\left(\sin\dfrac{\pi x}{N}\cos\pi x - \dfrac{1}{N}\sin\pi x\cos\dfrac{\pi x}{N}\right)\right), & -N<x<N, x\neq 0 \end{cases} \tag{C. 3a}$$

There are two subtractions here. The first one is the same as in the 1st-order derivative, for which we can use the same treatment as above. The second subtraction, $\sin^2\left(\frac{\pi x}{N}\right)\sin\pi x\left(-1+\frac{1}{N^2}\right)-\frac{2}{N}\cos\left(\frac{\pi x}{N}\right)f(\pi x)$, has the result on the same order of magnitude as the operands (in fact $\frac{2}{N}\cos\left(\frac{\pi x}{N}\right)f(\pi x)$ is about two thirds of $\sin^2\left(\frac{\pi x}{N}\right)\sin\pi x\left(-1+\frac{1}{N^2}\right)$), so there is no need to take special care of precision for this subtraction. The complete routine for calculating the 2nd-order derivative is

```
if $\sin(\pi x/N)=0, \mathrm{sinc}''_N(x)=-\frac{\pi^2}{3}\left(1-\frac{1}{N^2}\right)$;
else calculate $f(\pi x)$, then
    $\mathrm{sinc}''_N(x)=\frac{\pi^2}{N}\left(\frac{1-N^2}{N}\frac{\sin\pi x}{N\sin(\pi x/N)}-\frac{2\cos(\pi x/N)}{N\sin^3(\pi x/N)}f(\pi x)\right)$
```

When the 2nd-order derivative is computed along with the sinc function and the 1st-order derivative, previous results can be reused to calculate

$$\mathrm{sinc}''_N(x)=\frac{\pi^2}{N}\left(\frac{1-N^2}{N}\mathrm{sinc}_N(x)-\frac{2\cos(\pi x/N)}{N\sin^3(\pi x/N)}f(\pi x)\right). \qquad \text{(C. 3b)}$$

C.2 Computing λ(f) and its derivatives

$$\lambda(f)=\frac{<\mathbf{X},\mathbf{W}(f)>}{\|\mathbf{W}(f)\|^2} \qquad \text{(3. 5a)}^2$$

As mentioned in 3.1, it is of great help if the window DTFT $W(f)$ has a closed analytic form from which it can be directly calculated for any *f*. This is the case with the cosine window family, i.e. window functions in the form of $w_n=\sum_m(-1)^m c_m\exp^{j2\pi mn/N}$, with $c_m=c_{-m}$, $|m|\le M$. We can calculate *W*(*f*) as

$$W(f)=e^{-j\pi Nf}NH(f),\ H(f)=\sum_m c_m e^{-j\pi(m/N-f)}\mathrm{sinc}_N(m-Nf) \qquad \text{(A. 5e)}$$

Then

[2] Where we have omitted the caps "^" from λ and *f*.

$$\begin{aligned}\|\mathbf{W}(f)\|^2 &= <\mathbf{W}(f),\mathbf{W}(f)> = \sum_{k=k_1}^{k_2} W(k/N-f)W^*(k/N-f) \\ &= N^2\sum_{k=k_1}^{k_2}\left(\mathrm{Re}^2\,H(k/N-f)+\mathrm{Im}^2\,H(k/N-f)\right)\end{aligned} \tag{C. 4a}$$

$$\begin{aligned}<\mathbf{X},\mathbf{W}(f)> &= \sum_{k=k_1}^{k_2} X_k W^*(k/N-f) \\ &= e^{-j\pi Nf} N\sum_{k=k_1}^{k_2} X_k(-1)^k\left(\mathrm{Re}\,H(k/N-f)-j\,\mathrm{Im}\,H(k/N-f)\right)\end{aligned} \tag{C. 4b}$$

where

$$\begin{aligned}\mathrm{Re}\,H(k/N-f) &= \frac{(-1)^k\sin\pi Nf}{N}\sum_m(-1)^m c_m\cot\pi(\frac{m-k}{N}+f) \\ \mathrm{Im}\,H(k/N-f) &= -\frac{(-1)^k\sin\pi Nf}{N}\sum_m(-1)^m c_m\end{aligned} \tag{C. 4c}$$

Im$H(f)$ vanishes when $\sum_m(-1)^m c_m=0$, i.e. the window function has a vanishing moment not lower than the Hann window. To calculate $H(k/N-f)$, it is efficient to proceed with the index l=k-m, so that the triangular functions are called a minimal times.

During the LSE iterations, there is no need for the phase angle, as we are only interested in $|\lambda(f)|^2\|\mathbf{W}(f)\|^2$. Then there is no need to multiply the factor $e^{-j\pi Nf}$ for calculating <**X**, **W**(f)>.

The derivatives of $|\lambda(f)|^2$ are necessary for LSE searching. We have

$$|\lambda(f)|^2\|\mathbf{W}(f)\|^2=\frac{|<\mathbf{X},\mathbf{W}(f)>|^2}{\|\mathbf{W}(f)\|^2}\equiv\frac{F}{G} \tag{C. 5a}$$

It follows that

$$\frac{d|\lambda(f)|^2\|\mathbf{W}(f)\|^2}{df}=\frac{GF'-FG'}{G^2} \tag{C. 5b}$$

$$\frac{d^2|\lambda(f)|^2\|\mathbf{W}(f)\|^2}{df^2} = \frac{F''G^2 - 2F'GG' - FGG'' + 2FG'^2}{G^3} \tag{C. 5c}$$

where

$$F = \left|<\mathbf{X}, \mathbf{W}(f)>\right|^2 = \mathrm{Re}^2 <\mathbf{X}, \mathbf{W}(f)> + \mathrm{Im}^2 <\mathbf{X}, \mathbf{W}(f)> \tag{C. 5d}$$

$$F' = 2\,\mathrm{Re} <\mathbf{X}, \mathbf{W}(f)> \mathrm{Re}' <\mathbf{X}, \mathbf{W}(f)> + 2\,\mathrm{Im} <\mathbf{X}, \mathbf{W}(f)> \mathrm{Im}' <\mathbf{X}, \mathbf{W}(f)> \tag{C. 5e}$$

$$\begin{gathered} F'' = 2\,\mathrm{Re} <\mathbf{X}, \mathbf{W}(f)> \mathrm{Re}'' <\mathbf{X}, \mathbf{W}(f)> + 2\left(\mathrm{Re}' <\mathbf{X}, \mathbf{W}(f)>\right)^2 \\ + 2\,\mathrm{Im} <\mathbf{X}, \mathbf{W}(f)> \mathrm{Im}' <\mathbf{X}, \mathbf{W}(f)> + 2\left(\mathrm{Im}' <\mathbf{X}, \mathbf{W}(f)>\right)^2 \end{gathered} \tag{C. 5f}$$

$$G = \|\mathbf{W}(f)\|^2 = N^2 \sum_{k=k_1}^{k_2} \left(\mathrm{Re}^2 H(\frac{k}{N} - f) + \mathrm{Im}^2 H(k/N - f) \right) \tag{C. 5g}$$

$$G' = N^2 \sum_{k=k_1}^{k_2} \left(2\,\mathrm{Re}\, H(\frac{k}{N} - f)\,\mathrm{Re}'\, H(\frac{k}{N} - f) + 2\,\mathrm{Im}\, H(\frac{k}{N} - f)\,\mathrm{Im}'\, H(\frac{k}{N} - f) \right) \tag{C. 5h}$$

$$\begin{gathered} G'' = N^2 \sum_{k=k_1}^{k_2} \left(2\,\mathrm{Re}\, H(\frac{k}{N} - f)\,\mathrm{Re}''\, H(\frac{k}{N} - f) + 2\left(\mathrm{Re}'\, H(\frac{k}{N} - f) \right)^2 \right. \\ \left. + 2\,\mathrm{Im}\, H(\frac{k}{N} - f)\,\mathrm{Im}''\, H(\frac{k}{N} - f) + 2\left(\mathrm{Im}'\, H(\frac{k}{N} - f) \right)^2 \right) \end{gathered} \tag{C. 5i}$$

To calculate <**X**, **W**(*f*)> and its derivatives, we substitute $\widetilde{X}_k = (-1)^k X_k$, then an equivalent form of <**X**, **W**(*f*)> is

$$<\mathbf{X}, \mathbf{W}(f)> = N \sum_{k=k_1}^{k_2} \widetilde{X}_k \left(\mathrm{Re}\, H(\frac{k}{N} - f) - j\,\mathrm{Im}\, H(\frac{k}{N} - f) \right), \tag{C. 5j}$$

so

$$\mathrm{Re} <\mathbf{X}, \mathbf{W}(f)> = N \sum_{k=k_1}^{k_2} \mathrm{Re}\, \widetilde{X}_k\; \mathrm{Re}\, H(k/N - f) + \mathrm{Im}\, \widetilde{X}_k\; \mathrm{Im}\, H(k/N - f), \tag{C. 5k}$$

$$\mathrm{Re}^{(n)} <\mathbf{X}, \mathbf{W}(f)> = N \sum_{k=k_1}^{k_2} \mathrm{Re}\, \widetilde{X}_k\; \mathrm{Re}^{(n)}\, H(k/N - f) + \mathrm{Im}\, \widetilde{X}_k\; \mathrm{Im}^{(n)}\, H(k/N - f), \tag{C. 5l}$$

$$\operatorname{Im} < \mathbf{X}, \mathbf{W}(f) >= N \sum_{k=k_1}^{k_2} \operatorname{Im} \widetilde{X}_k \operatorname{Re} H(k/N - f) - \operatorname{Re} \widetilde{X}_k \operatorname{Im} H(k/N - f), \qquad \text{(C. 5m)}$$

$$\operatorname{Im}^{(n)} < \mathbf{X}, \mathbf{W}(f) >= N \sum_{k=k_1}^{k_2} \operatorname{Im} \widetilde{X}_k \operatorname{Re}^{(n)} H(k/N - f) - \operatorname{Re} \widetilde{X}_k \operatorname{Im}^{(n)} H(k/N - f).$$

$$\text{(C. 5n)}$$

The derivatives of Re $H(k/\mathrm{N}\text{-}f)$ are calculated by

$$\begin{aligned} \operatorname{Re}' H(k/N - f) &= (-1)^k \pi \cos \pi N f \sum_m (-1)^m c_m \cot \pi (\frac{m-k}{N} + f) \\ &\quad - \frac{(-1)^k \pi \sin \pi N f}{N} \sum_m (-1)^m c_m \csc^2 \pi (\frac{m-k}{N} + f) \end{aligned}, \qquad \text{(C. 5o)}$$

$$\begin{aligned} \operatorname{Re}'' H(k/N - f) &= -(-1)^k \pi^2 N \sin \pi N f \sum_m (-1)^m c_m \cot \pi (\frac{m-k}{N} + f) \\ &\quad - 2(-1)^k \pi^2 \cos \pi N f \sum_m (-1)^m c_m \csc^2 \pi (\frac{m-k}{N} + f) \qquad \text{(C. 5p)} \\ &\quad + \frac{2(-1)^k \pi^2 \sin \pi N f}{N} \sum_m (-1)^m c_m \pi \csc^2 \pi (\frac{m-k}{N} + f) \cot \pi (\frac{m-k}{N} + f) \end{aligned}$$

or

$$\begin{aligned} \operatorname{Re}' H(k/N - f) &= \sum_m c_m \Big(- \pi \sin \pi (\frac{m-k}{N} + f) \operatorname{sinc}_N (m - k + Nf) \\ &\quad + N \cos \pi (\frac{m-k}{N} + f) \operatorname{sinc}'_N (m - k + Nf) \Big) \end{aligned} \qquad \text{(C. 5q)}$$

$$\begin{aligned} \operatorname{Re}'' H(k/N - f) &= \sum_m c_m \Big(- \pi^2 \cos \pi (\frac{m-k}{N} + f) \operatorname{sinc}_N (m - k + Nf) \\ &\quad - 2N\pi \sin \pi (\frac{m-k}{N} + f) \operatorname{sinc}'_N (m - k + Nf) \qquad \text{(C. 5r)} \\ &\quad + N^2 \cos \pi (\frac{m-k}{N} + f) \operatorname{sinc}''_N (m - k + Nf) \Big) \end{aligned}$$

The first set of computations (C.5o) and (C.5p) is faster, but the second set (C.5q) and (C.5r) is safer when any of the angles is close to zero. The derivatives of Im $H(k/\mathrm{N}\text{-}f)$ are calculate by

$$\mathrm{Im}' H(k/N-f) = -(-1)^k \pi \cos \pi N f \sum_m (-1)^m c_m \tag{C.5s}$$

$$\mathrm{Im}'' H(k/N-f) = (-1)^k \pi^2 N \sin \pi N f \sum_m (-1)^m c_m \tag{C.5t}$$

When Im$H(f)$ vanishes for the window function, so do its derivatives.

C.3 Using LSE method on non-stationary sinusoids

We apply the LSE estimator, in its simplified version, on time-varying sinusoids. Let our sinusoid be

$$x_n = a_n e^{j(\varphi_c + 2\pi \int_{N/2}^{n} f(t)dt)}, \tag{3. 9a}$$

and the symmetric window function be w. Define $\varphi_{mn} = 2\pi \int_m^n f(t)dt$, $\varphi_n = \varphi_{mn}\big|_{m=N/2}$. The short-time Fourier transform of the sinusoid is

$$X_k = \sum_{n=0}^{N-1} w_n a_n e^{j(\varphi_c + \varphi_n - 2\pi k n/N)} \tag{C. 6a}$$

The window spectrum is

$$W_k(f) = W(k/N-f) = \sum_{n=0}^{N-1} w_n e^{-j2\pi(k/N-f)n} \tag{C. 6b}$$

We compute the inner product on the whole frequency range:

$$<\mathbf{X}, \mathbf{W}(f)> = \sum_{k=0}^{N-1} \sum_{n=0}^{N-1} w_n a_n e^{j(\varphi_c + \varphi_n)} \cdot e^{-j2\pi k n/N} \sum_{m=0}^{N-1} w_m e^{-j2\pi f m} \cdot e^{j2\pi k m/N} \tag{C. 7a}$$

Since

$$\sum_{k=0}^{N-1} \exp(-j2\pi k(n-m)/N) = N\delta(n-m), \tag{C. 7b}$$

we have

$$<\mathbf{X},\mathbf{W}(f)>=N\sum_{n} w_n^2 a_n e^{j(\varphi_c+\varphi_n)} e^{-j2\pi fn} \tag{C. 7c}$$

or

$$<\mathbf{X},\mathbf{W}(f)>=Ne^{j\varphi_c}\sum_{n} b_n e^{j(\varphi_n-j2\pi fn)}\ ,\ b_n \equiv w_n^2 a_n\,. \tag{C. 7d}$$

Define

$$\Delta\varphi_{mn}=\varphi_{mn}-2\pi(n-m)f\,. \tag{C. 8}$$

The square norm of the inner product is

$$\left\|<\mathbf{X},\mathbf{W}(f)>\right\|^2 = N^2\sum_{n=0}^{N-1}\sum_{m=0}^{N-1} b_m b_m \cos\Delta\varphi_{mn} \tag{C. 9a}$$

To maximize the above we set its derivative regarding *f* to 0:

$$\frac{d}{df}\left\|<\mathbf{X},\mathbf{W}(f)>\right\|^2 = N^2\cdot 2\pi\sum_{n=0}^{N-1}\sum_{m=0}^{N-1}(n-m)b_n b_m \sin\Delta\varphi_{mn}=0 \tag{C. 9b}$$

Define

$$w_{mn}=(n\text{-}m)w_m{}^2 w_n{}^2, \tag{C. 10}$$

then we have

$$\hat{f}=\frac{\displaystyle\sum_{n=0}^{N-1}\sum_{m=0}^{N-1} w_{mn}a_n a_m \int_m^n f(t)dt\,\mathrm{sinc}\frac{\Delta\varphi_{mn}}{\pi}}{\displaystyle\sum_{n=0}^{N-1}\sum_{m=0}^{N-1} w_{mn}a_n a_m (n-m)\,\mathrm{sinc}\frac{\Delta\varphi_{mn}}{\pi}}\,, \tag{3. 9b}$$

where sinc(*x*) is the continuous sinc function. ■

We study the amplitude and frequency symmetry as follows. Let

$$a_n^e=\frac{a_n+a_{N-n}}{2}\,,\ a_n^0=\frac{a_n-a_{N-n}}{2}\,, \tag{C. 11a}$$

$$f^e(t)=\frac{f(t)+f(N-t)}{2}\,,\ f^o(t)=\frac{f(t)-f(N-t)}{2}\,. \tag{C. 11b}$$

Then

$$a_n = a_n^e + a_n^o \,,\; f(t) = f^e(t) + f^o(t)\,, \tag{C. 11c}$$

and

$$a_n^e = a_{N-n}^e \,,\; a_n^o = -a_{N-n}^o \,,\; f^e(t) = f^e(N-t)\,,\; f^o(t) = -f^o(N-t) \tag{C. 11d}$$

We rewrite (C.9b) as

$$\sum_{n=0}^{N-1}\sum_{m=0}^{N-1} y_{mn} = 0\,,\; y_{mn} = w_{mn} a_n a_m \sin\Delta\varphi_{mn} \tag{C. 12}$$

Define

$$\varphi_{mn}^d = 2\pi\int_m^n (f_e(t) - f)dt\,,\; \varphi_{mn}^o = 2\pi\int_m^n f_o(t)dt\,, \tag{C. 13a}$$

$$s_* = \sin\varphi_*\,,\; c_* = \cos\varphi_*\,, \tag{C. 13b}$$

where * stands for subscripts and superscripts such as *mn*, etc., and

$$a_{mn}^{e2} = a_m^e a_n^e + a_m^o a_n^o \,,\; a_{mn}^{o2} = a_m^e a_n^o + a_m^o a_n^e \,. \tag{C. 13c}$$

It’s trivial to show that

$$y_{mn} = w_{mn}(a_{mn}^{e2} + a_{mn}^{o2})(s_{mn}^d c_{mn}^o + c_{mn}^d s_{mn}^o)\,,$$

$$y_{N-m,N-n} = w_{mn}(a_{mn}^{e2} - a_{mn}^{o2})(s_{mn}^d c_{mn}^o - c_{mn}^d s_{mn}^o) \tag{C. 14a}$$

Since y_{nn}=0, $\forall\, n$, we rewrite the left hand side of (C.12) as

$$\sum_{n=0}^{N-1}\sum_{m=0}^{N-1} y_{mn} = \sum_{n=1}^{N-1}\sum_{m=0}^{n-1}(y_{mn} + y_{N-m,N-n})\,. \tag{C. 14b}$$

Add the two equations in (C.14a) we get

$$y_{mn} + y_{N-m,N-n} = 2w_{mn}(a_{mn}^{e2} s_{mn}^d c_{mn}^o + a_{mn}^{o2} s_{mn}^o c_{mn}^d) \tag{C. 14c}$$

Therefore

$$\sum_{n=1}^{N-1}\sum_{m=0}^{n-1}(y_{mn} + y_{N-m,N-n}) = 2\sum_{n=1}^{N-1}\sum_{m=0}^{n-1} w_{mn}(a_{mn}^{e2} s_{mn}^d c_{mn}^o + a_{mn}^{o2} s_{mn}^o c_{mn}^d) = 0 \tag{C. 14d}$$

Using the same technique as in deriving (3.9c) (see 3.1.3), we get

$$\hat{f} = \frac{\sum_{l=0}^{N-2}\left(\eta_l^e \int_0^1 f^e(l+t)dt + \eta_l^o \int_0^1 f^o(l+t)dt\right)}{\sum_{l=0}^{N-2}\eta_l^e}, \qquad \text{(C. 15a)}$$

where

$$\eta_l^e = \sum_{n=l+1}^{N-1}\sum_{m=0}^{l} w_{mn} a_{mn}^{e2} \operatorname{sinc}\frac{\varphi_{mn}^d}{\pi}\cos\varphi_{mn}^o,\ \eta_l^o = \sum_{n=l+1}^{N-1}\sum_{m=0}^{l} w_{mn} a_{mn}^{o2} \sin\frac{\varphi_{mn}^o}{\pi}\cos\varphi_{mn}^d. \quad \text{(C. 15b)}$$

If the estimate is assigned to the window centre, then the frequency estimate error is

$$\hat{f} - f(N/2) = \frac{\sum_{l=0}^{N-2}\left(\eta_l^e \int_0^1 (f^e(l+t) - f(N/2))dt + \eta_l^o \int_0^1 f^o(l+t)dt\right)}{\sum_{l=0}^{N-2}\eta_l^e}. \qquad \text{(C. 15c)}$$

We make the following observations regarding the results.

1. In the context of slow parameter variation, the even parts of the parameters have much higher amplitude level than the odd parts.

2. a_{mn}^{e2} and a_{mn}^{o2}, in general, represent the even and odd parts of the amplitude respectively, while φ_{mn}^d and φ_{mn}^o represents the even and odd parts of the frequency. As a result η_l^e is roughly associated with the even part, and η_l^o with the odd part, of the amplitude. η_l^o vanishes if the amplitude is even-symmetric.

3. (C.15a) shows that the odd part of the frequency contributes to the estimate only if the amplitude has an odd part too. In other words, if the frequency is odd-symmetric (subject to a constant shift) and the amplitude is even-symmetric, then the LSE estimator does not incur a frequency estimation error. In particular, the LSE estimator is accurate in measuring the instantaneous frequency of a linear chirp.

Appendix D

Calculations in the harmonic particle detector

D.1 Solving inequality system (3.17)

Let $g_{m-}=\left(\frac{\hat{f}^m-\Delta^m}{m}\right)^2$, $g_{m+}=\left(\frac{\hat{f}^m+\Delta^m}{m}\right)^2$, $k=m^2-1$, then the inequality system (3.17) becomes

$$\begin{cases} g_{m_1-}<F+k_1G<g_{m_1+} \\ g_{m_2-}<F+k_2G<g_{m_2+} \\ \qquad\qquad\cdots\cdots \\ g_{m_M-}<F+k_MG<g_{m_M+} \end{cases} \tag{D. 1}$$

In practice there is always a reasonable range for $f^1=(F)^{1/2}$ and $B=G/F$. f^1 can never be below 0 or above 1/2 (or N/2 bins, where N is the size of the DFT), and the stiffness coefficient B is always positive and below some B_M<<1. Let a pre-determined range of F be $g_{0-}<F<g_{0+}$, then we can initial the range R by

$$\begin{cases} g_{0-}<F<g_{0+} \\ 0\le G<B_MF \end{cases} \tag{D. 2}$$

This is shown in Figure D.1.

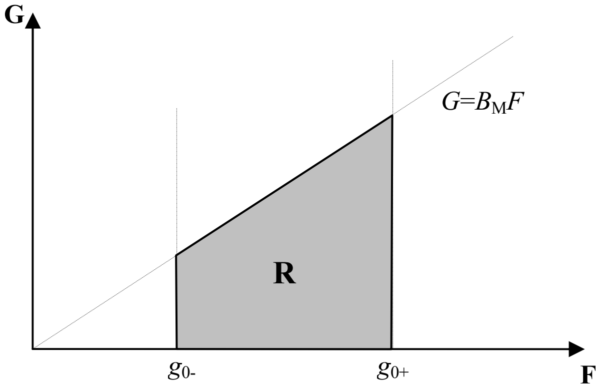


Figure D. 1 Preset range of R in F-G plane

We proceed with the M inequalities one by one. The solution of $g_{m-} < F + kG < g_{m+}$ is a stripe in the F-G plane that falls between the two parallel lines $F + kG = g_{m-}$ and $F + kG = g_{m+}$. So for each inequality of (D.1), we use these two lines to cut off the part of R that lies outside them. Accordingly R becomes smaller and smaller as more and more partials are added. If at any stage R becomes empty, then the solution set of (D.1) is empty. Throughout this process R is a convex polygon.

We represent R by its N vertices in the format $\{N;\ (F_n,\ G_n)_{0\le n<N}\}$, arranged in clockwise order and starting from the leftmost (smallest F) vertex. In the case of two leftmost vertices, i.e. there being a left most side normal to the F axis, we arrange the vertex list to start from the upper one (larger G) and end at the lower one. For example, the range R in Figure D.1 is represented as $\{4;\ (F_n,\ G_n)_{n=0,1,2,3}\}$, with $F_0=F_3=g_{0-}$, $F_1=F_2=g_{0+}$, $G_0=B_Mg_{0-}$, $G_1=B_Mg_{0+}$, $G_2=G_3=0$.

With R initialized as $\{N;\ (F_n,\ G_n)_{0\le n<N}\}$, now we wish to apply the constraints $F + kG < g_{m+}$ and $F + kG > g_{m-}$ to R. To apply any linear constraint $aF+bG+c<0$, we collect all the vertices that satisfy the constraint. There are three possibilities:

1. no vertex satisfies it, then R is reduced to Φ;
2. all vertices satisfy it, then the constraint does not change R at all;
3. some vertices satisfy it and some do not, then since R is a vertex polygon and $aF+bG+c<0$ is a linear constraint, all vertices that satisfy the constraint adhere together and so do the rest. There are two possibilities:

3.1. both (F_0, G_0) and ($F_{N\text{-}1}$, $G_{N\text{-}1}$) satisfy the constraint, then there exist $n_s \leq N$-1 and $n_e \geq 1$, $n_e < n_s$, so that vertices n_s, n_s+1, …, N-1, 0, …, n_e-1 satisfy the constraint, and vertices n_e, …, n_s-1 do not. We calculate the intersection of line l: $aF+bG+c=0$ with the line segment connecting vertices n_s-1 and n_s, name it (F_s, G_s), and with the line segment connecting vertices n_e-1 and n_e, name it (F_e, G_e), then update the vertex sequence of R as

$$0, 1, ..., n_e\text{-}1, e, s, n_s, …, N\text{-}1.$$

3.2. either (F_0, G_0) or ($F_{N\text{-}1}$, $G_{N\text{-}1}$) does not satisfy the constraint, then there exist $0 \leq n_s < n_e \leq N$, so that vertices n_s, n_s+1, …, n_e-1 satisfy the constraint, and vertices 0, …, n_s-1, (this part does not exist if n_s=0) and n_e, …, N-1 (this part does not exist if n_e=N) do not. We calculate the intersection of line l with the line segment connecting vertices n_s-1 and n_s (or N-1 and 0 if n_s=0), name it (F_s, G_s), and with the line segment connecting vertices n_e-1 and n_e (or N-1 and 0 if n_e=N), name it (F_e, G_e), then update the vertex sequence of R as

$$0, 1, …, n_e\text{-}1, e, s, \quad \text{if } n_s=0, \text{ or}$$

$$s, n_s, n_s+1, …, n_e\text{-}1, e, \quad \text{if } n_s \neq 0 \text{ and } F_s < F_e, \text{ or}$$

$$e, s, n_s, n_s+1, …, n_e\text{-}1, \quad \text{if } n_s \neq 0 \text{ and } F_s > F_e$$

All the three sequences are clockwise. Now we show that their leftmost vertices are 0, s, and e respectively. The first case if obvious, since vertex 0 is the leftmost point of R. When $n_s \neq 0$, then vertex 0 may fall on l, on the left side of l, or on the right side of l. First let it be on l, then either n_s=1 (so s=0) or n_e=N (so e=0), in both cases vertex 0, the leftmost point of R, is one of s or e. Then we let vertex 0 be on the left side of l, so the vertices s, n_s, n_s+1, …, n_e-1, e are not on the left side (Figure D.2). Each of the vertices, say (F_n, G_n), casts an image, say (F'_n, G'_n), on line l, where the line between it and vertex 0 crosses l. And since these vertices come clockwise regarding vertex 0, so do their image points. Noticing the slope of l is always negative, these clockwise image points come in left-to-right order, so $F_s = F'_s < F'_n \leq \mathrm{F}_n$, i.e. s is the leftmost vertex. Similarly, if vertex 0 is on the right side of l, then e s the leftmost vertex. In any of the three cases, the leftmost point is either s

or *e*, and can be found by comparing F_s and F_e. In the rare case that F_s=F_e, i.e. adding a fundamental frequency to R, we further compare G_s and G_e.

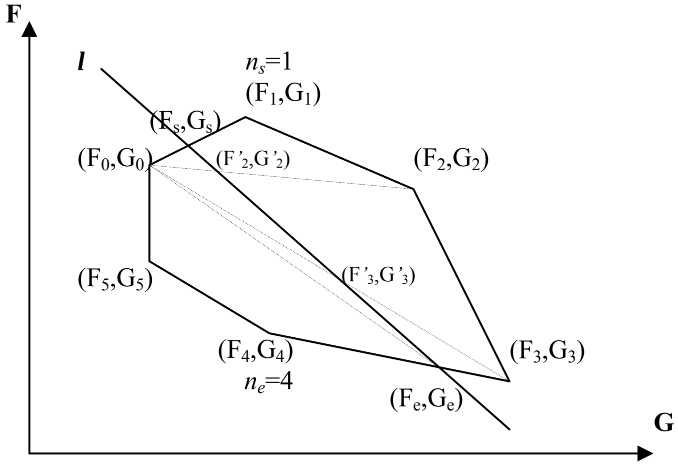


Figure D. 2 Cutting R with linear constraints

D.2 Minimal-maximum search for (3.26c)

$$(\hat{F},\hat{G}) = \arg \inf_{(F,G)\in R} \max_{m} \frac{\left|\hat{f}^{m} - m\sqrt{F+(m^2-1)G}\right|}{\Delta^m} \qquad (3.\ 26c)$$

We remove the absolute value operator by mirroring the range of *m*:

$$(\hat{F},\hat{G}) = \arg \inf_{(F,G)\in R} \max_{m} \left\{ \frac{m\sqrt{F+(m^2-1)G} - \hat{f}^{m}}{\Delta^m}, \frac{-m\sqrt{F+(m^2-1)G} + \hat{f}^{m}}{\Delta^m} \right\} \qquad (D.\ 3)$$

$$\equiv \arg \inf_{(F,G)\in R} \theta(F,G)$$

The maximum is taken over 2*M* functions. To make it explicit, we define k_l=m_l^2-1, $e_{2l}(F,G) = \frac{m_l\sqrt{F+k_lG} - \hat{f}^{m_l}}{\Delta^{m_l}}$, $e_{2l+1}(F,G) = \frac{-m_l\sqrt{F+k_lG} + \hat{f}^{m_l}}{\Delta^{m_l}}$, *l*=0, 1, …, *M*-1, then (D.3) becomes

$$(\hat{F},\hat{G}) = \arg \inf_{(F,G)\in R} \max_{0\le l<2M} e_l(F,G)\,,\ \theta(F,G) = \max_{0\le l<2M} e_l(F,G) \qquad (D.\ 4)$$

We use the symbol $(\hat{F}_n,\hat{G}_n)_n$ to refer to a sequence of points in the *F*-*G* plane, indexed on *n*. Any point (*F*, *G*) inside R must have *θ*(*F*, *G*)<1; any point (*F*, *G*) on one side of R must have $\theta(F, G)=1$, except the two sides given by $G=0$ and $G=B_M F$. To find the minimal maximum, we compute a sequence of points $(\hat{F}_n, \hat{G}_n)$, $n=1, 2, \ldots$, so that they all fall on R and $\theta(\hat{F}_n, \hat{G}_n)$ decreases with n, until at some point we reach the minimum.

We start from a starting point $(\hat{F}_0, \hat{G}_0) \in$R. This can be conveniently taken at any vertex. In the n^{th} step, $n=1, 2, \ldots$, we move from $(\hat{F}_{n-1}, \hat{G}_{n-1})$ to $(\hat{F}_n, \hat{G}_n)$, so that $\theta(\hat{F}_n, \hat{G}_n) < \theta(\hat{F}_{n-1}, \hat{G}_{n-1})$. Regarding each step, we denote the starting position as ($F0$, $G0$) and end position as ($F1$, $G1$). The polygon R is described with a vertex list $\{N; (F_n, G_n)_{0\leq n<N}\}$, where the vertices are arranged in clock-wise order. Let $\mathbf{e}_F$ and $\mathbf{e}_G$ be unit vectors in the F and G direction respectively, and the vector pointing from the (F_n, G_n) to the next vertex in the list be $\mathbf{r}_n$. Then

$$\mathbf{r}_n = \begin{cases} (F_{n+1} - F_n)\mathbf{e}_F + (G_{n+1} - G_n)\mathbf{e}_G, & n = 0, 1, \cdots, N-1 \\ (F_0 - F_n)\mathbf{e}_F + (G_0 - G_n)\mathbf{e}_G, & n = N-1 \end{cases} \qquad \text{(D. 5a)}$$

We also refer to the side of R starting from (F_n, G_n) as $\mathbf{r}_n$. The side of R that comes before $\mathbf{r}_n$, i.e. the side that ends at (F_n, G_n), is denoted as $\mathbf{r}_{n-}$. Obviously

$$\mathbf{r}_{n-} = \begin{cases} \mathbf{r}_{n-1}, & n = 1, \cdots, N-1 \\ \mathbf{r}_{N-1}, & n = 0 \end{cases} \qquad \text{(D. 5b)}$$

The following results will be useful in determining the minimal-maximum searching directions:

1. A vector $\mathbf{r}$ starting from a point on side $\mathbf{r}_n$ points inside R if and only if $(\mathbf{r}\times\mathbf{r}_n)\cdot(\mathbf{e}_F\times\mathbf{e}_G)>0$;

2. A vector $\mathbf{r}$ starting from the vertex (F_n, G_n) points inside R if and only if $(\mathbf{r}\times\mathbf{r}_n)\cdot(\mathbf{e}_F\times\mathbf{e}_G)>0$ and $(\mathbf{r}\times\mathbf{r}_{n-})\cdot(\mathbf{e}_F\times\mathbf{e}_G)>0$;

3. A point (F, G) is on the R side of $\mathbf{r}_n$ if and only if $(F-F_n)\mathbf{e}_F+(G-G_n)\mathbf{e}_G$ points inside R;

4. A point (F, G) is inside the polygon R if (F, G) is on the R of any $\mathbf{r}_n$, n=1, …, N-1.

D.2.1 Local and global minimal maximum

We first study the local minimal maximum. Let (F, G) be a point in the F-G plane, and $O_r(F, G)$ be an r-vicinity of point (F, G) within polygon R, i.e.

$$O_r(F, G)=\{(f, g)|(f, g)\in \mathrm{R}, (f-F)^2+(g-G)^2<r^2\}. \qquad \text{(D. 6a)}$$

We say that (F, G) is a local minimal maximum of $\{e_l\}_{0\leq l<2M}$, if $\exists r>0$, so that $\forall(f, g)\in O_r(F, G)$, $\theta(f, g)\geq\theta(F, G)$. When r is small enough, the shape of $O_r(F, G)$ depends on the position of (F, G) in R. If (F, G) is inside R, $O_r(F, G)$ is a circle; if (F, G) lies on one side of R excluding the two ends, $O_r(F, G)$ becomes half circle; and if (F, G) is a vertex of R, then $O_r(F, G)$ is a sector, with the circular centre at (F, G). The shape of $O_r(F, G)$ determines the feasible searching directions from point (F, G). We summarize these directions by defining

$$D_{\mathrm{R}}(F,G)=\left\{\frac{(f-F)\mathbf{e}_F+(g-G)\mathbf{e}_G}{\left((f-F)^2+(g-G)^2\right)^{1/2}}\middle|(f,g)\in R\right\}. \qquad \text{(D. 6b)}$$

I we express a vector $f\mathbf{e}_F+g\mathbf{e}_G$ in short form (f, g), we get

$$D_{\mathrm{R}}(F,G)=\left\{\frac{(f-F,\quad g-G)}{\left((f-F)^2+(g-G)^2\right)^{1/2}}\middle|(f,g)\in R\right\}. \qquad \text{(D. 6c)}$$

$D_{\mathrm{R}}(F, G)$ is the collection of unit vectors pointing from (F, G) toward anywhere inside R. Since R is convex, $D_{\mathrm{R}}(F, G)$ gives the feasible searching directions at (F, G). For a small enough r, $(f, g)\in O_r(F, G)$ if and only if there exists $0\leq r_0<r$ and $(\delta F,\delta G)\in D_{\mathrm{R}}(F,G)$, so that $(f, g)=(F, G)+r_0(\delta F, \delta G)$. The following proposition gives the condition for a point (F, G) to be a local minimal maximum.

Lemma : $\nabla e_l(F, G)$ has constant direction, $0\leq l<2M-1$.

To show this we calculate $\nabla e_{2l} = \dfrac{m_l(\mathbf{e}_F + k_l\mathbf{e}_G)}{2\Delta^{m_l}\sqrt{F + k_l G}} = \dfrac{m_l^{\,2}}{2\Delta^{m_l}\left(e_{2l}\Delta^{m_l} + \hat{f}^{m_l}\right)}\left(\mathbf{e}_F + k_l\mathbf{e}_G\right)$,

and $\nabla e_{2l+1} = -\dfrac{m_l(\mathbf{e}_F + k_l\mathbf{e}_G)}{2\Delta^{m_l}\sqrt{F + k_l G}} = \dfrac{m_l^{\,2}}{2\Delta^{m_l}\left(e_{2l+1}\Delta^{m_l} - \hat{f}^{m_l}\right)}\left(\mathbf{e}_F + k_l\mathbf{e}_G\right)$. ■

Proposition D.1: Let $e_{l_1}(F,G) = e_{l_2}(F,G) = \ldots = e_{l_K}(F,G) = \theta(F, G)$. (F, G) is a local minimal maximum if and only if $\forall\, \mathbf{e} \in D_{\mathrm{R}}(F, G)$, there exists k, $1 \le k \le K$, so that $\nabla e_{l_k} \cdot \mathbf{e} \ge 0$.

First let (F, G) be a local minimal maximum, and $\mathbf{e} \in D_{\mathrm{R}}(F, G)$. As all e_l's are continuous with bounded gradients, there exists $r_1 > 0$ so that one of e_{l_1}, …, e_{l_K} remains being the maximal e_l within $O_{r_1}(F,G)$. Consider $(f, g) = (F, G) + r_0\mathbf{e}$, $0 < r_0 < \min(r, r_1)$. Let the maximal e_l at (f, g) be e_{l_k}. According to the definition of minimal maximum, $e_{l_k}(F, G) \le e_{l_k}(f, g)$. As $\nabla \mathbf{e}_{l_k}$ has constant direction, $\nabla e_{l_k} \cdot \mathbf{e} \ge 0$. To show the inverse, let (F, G) be not a local minimal maximum, so that for any $0 < r_0 < r$, there exists $(f, g) \in O_{r_1}(F,G)$ so that $\max_l e_l(f, g) < \max_l e_l(F, G)$, that is, $e_{l_1}(f, g) < e_{l_1}(F, G)$, $e_{l_2}(f, g) < e_{l_2}(F, G)$, …, $e_{l_K}(f, g) < e_{l_K}(F, G)$. Define $\mathbf{e} = \dfrac{(f-F)\mathbf{e}_F + (g-G)\mathbf{e}_G}{\left((f-F)^2 + (g-G)^2\right)^{1/2}} \in D_{\mathrm{R}}(F, G)$, then for $\forall\, k$, $1 \le k \le K$, $\nabla e_{l_k} \cdot \mathbf{e} < 0$, since the gradients $\nabla \mathbf{e}_{l_k}$ have constant directions. This concludes the proof of the proposition.

■

The following proposition shows that the local minimal maximum is unique.

Proposition D.2: A local minimal maximum in R is the minimal maximum in R.

Let (F, G) be a local minimal maximum in R, $e_{l_1}(F,G) = e_{l_2}(F,G) = \ldots = e_{l_K}(F,G) = \theta(F, G)$. Suppose there is another $(f, g) \in \mathrm{R}$, so that $\max_l e_l(f, g) < \max_l e_l(F, G)$. that is, $e_{l_1}(f, g) < e_{l_1}(F, G)$, $e_{l_2}(f, g) < e_{l_2}(F, G)$, …, $e_{l_K}(f, g) < e_{l_K}(F, G)$. Define $\mathbf{e} = \dfrac{(f-F)\mathbf{e}_F + (g-G)\mathbf{e}_G}{\left((f-F)^2 + (g-G)^2\right)^{1/2}} \in D_{\mathrm{R}}(F, G)$. Then $\nabla e_{l_k} \cdot \mathbf{e} < 0$ for $\forall\, k$, $1 \le k \le K$, because the gradients have constant directions. Using proposition D.1, (F, G)

can not be a local minimal maximum, which contradicts our assumption. Therefore such an (*f*, *g*) does not exist, so (*F*, *G*) must be the minimal maximum in R. ■

These two propositions make sure that we only need to find a local minimal maximum that satisfies the condition in Proposition D.1 for solving (3.26c). In each step, we determine how to find (*F*1, *G*1) according to the position of (*F*0, *G*0) and the number of equal maxima at (*F*0, *G*0).

D.2.2 Conditions of (*F*0,*G*0)

There are three different positions of (*F*0, *G*0), i.e. inside R, on a side of R, or at a vertex of R. At each position there are three different conditions for $\max_l e_l$, i.e. a solo maximum, two equal maxima, and more than two equal maxima.

D.2.2.1 (F0, G0) being inside R.

When (*F*0, *G*0) is inside R, all directions are feasible as the searching direction.

(1a) There are more than two equal maxima at (*F*0, *G*0). Let the equal maxima be $e_{l_1}(F0,G0)=e_{l_2}(F0,G0)=\ldots=e_{l_K}(F0,G0)$, *K*>2. The following proposition shows that there exist *l*1, *l*2∈{l_k|*k*=1, …, *K*}, so that down the decreasing direction of curve $e_{l1}=e_{l2}$, e_{l1} and e_{l2} remain maximal of the 2*M* relative errors.

Proposition D.3: if (*F*0, *G*0)∈R is not a minimal maximum, and $e_1=e_2=\ldots=e_K$ are *K* equal maxima at (*F*0, *G*0), *K*>2, then there exist *l*1 and *l*2, 1≤*l*1*,* *l*2≤*K*, so that ∀ 1≤*k*≤K, along the decreasing direction of $e_{l1}=e_{l2}$, e_k - e_{l1} is non-increasing.

Since (*F*0, *G*0) is not a minimal maximum, there exists **e**∈D_{R}(*F*0, *G*0), so that ∀ 1≤*k*≤*K*, $\nabla e_k\cdot\mathbf{e}<0$. This in turn implies that if we put these gradients in the *F*-*G* plane, the set of gradient points is linearly separable from the origin. Accordingly, the convex hull, let it be H, of all the gradient points does not contain the origin. Then there exists one side of H, let it be *h*, that separates H from the origin. Let its ends be ∇e_{l1} and ∇e_{l2}.

The increasing direction of the curve $e_{l1}=e_{l2}$ is given as

$$\delta_{12} = \frac{\nabla_G(e_{l1}-e_{l2})\mathbf{e}_F - \nabla_F(e_{l1}-e_{l2})\mathbf{e}_G}{\nabla_F e_{l2}\nabla_G e_{l1} - \nabla_F e_{l1}\nabla_G e_{l2}}, \text{ or } \delta_{12} = \frac{(\mathbf{e}_F\nabla_G - \mathbf{e}_G\nabla_F)(e_{l1}-e_{l2})}{\nabla_F e_2\nabla_G e_1 - \nabla_F e_1\nabla_G e_2}, \qquad \text{(D. 7)}$$

where the partial operators $\nabla_F = \partial/\partial F$, $\nabla_G = \partial/\partial G$. This vector is normalized so that $\delta_{12}\cdot\nabla e_{l1}=\delta_{12}\cdot\nabla e_{l2}=1$. An unnormalized version is given as

$$\nabla_{12} = \frac{(\nabla_F e_{l2}\nabla_G e_{l1} - \nabla_F e_{l1}\nabla_G e_{l2})\cdot(\nabla_G(e_{l1}-e_{l2})\mathbf{e}_F - \nabla_F(e_{l1}-e_{l2})\mathbf{e}_G)}{\left\|\nabla(e_{l1}-e_{l2})\right\|^2},$$

$$\text{or } \nabla_{12} = \frac{\left\|\nabla e_{l1}\times\nabla e_{l2}\right\|^2}{\left\|\nabla(e_{l1}-e_{l2})\right\|^2}\delta_{12}. \qquad \text{(D. 8)}$$

It is easy to show that $\delta_{12}\cdot\nabla_{12}=1$. ∇_{12} is interpreted as the altitude vector of the triangle (O, ∇e_{l1}, ∇e_{l2}) from the origin O. It is then straightforward to verify that along the opposite direction of ∇_{12}, e_k-e_1 is always non-increasing, because $\nabla(e_k-e_{l1})$ always points to the same side of h as ∇_{12}, perpendicular h (Figure D.3). ■

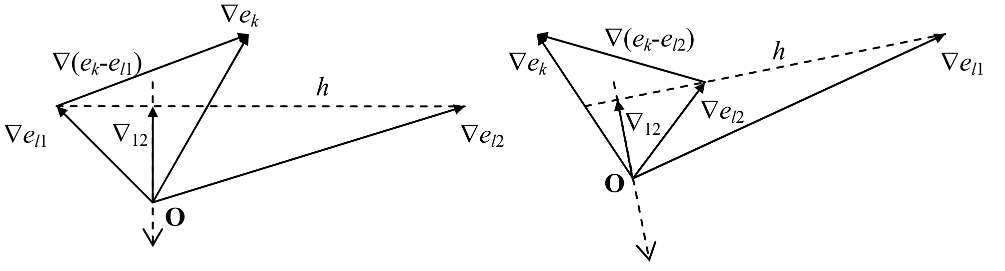


Figure D. 3 Geometric definition of ∇_{12}

We define the pair $l1$ and $l2$ that satisfy proposition D.3 an *equal-maximum pair*. This proof not only shows that there exists such a pair $l1$ and $l2$, but also provides a way to find them. In most cases K=3, then it is easy to test the signs of $\nabla_{12}\cdot\nabla(e_{l3}-e_{l1})$, $\nabla_{13}\cdot\nabla(e_{l2}-e_{l1})$, and $\nabla_{23}\cdot\nabla(e_{l1}-e_{l2})$. If all the three are negative, then (F0, G0) is the minimal maximum. If there is one being positive, say $\nabla_{pq}\cdot\nabla(e_r-e_p)$, then we can set $l1=p$ and $l2=q$. If there are two being positive, we choose the faster decreasing pair by picking the one with the larger $|\nabla_{pq}|$. In the rare case of K>3, we use the direct method given in Proposition D.3. However, we must first find the side of H that separates H from the origin. To do this we first find a direction **r** in which all the K maxima increase. This is done by collecting the direction angles of the K gradients ∇e_{l_k} into a smaller-than-π range. If this is impossible, then ($F0$, $G0$) is a minimal maximum already. Otherwise, denote the smallest angle θ_1, the largest $\theta_2<\theta_1+\pi$. Any direction within range ($\theta_2+\pi/2$, $\theta_1+3\pi/2$) is a decreasing direction of all the K errors. We can choose –**r** to be any vector with a direction angle in this range, and choose $l1$ by picking out the smallest $\nabla e_k \cdot \mathbf{r}$. It is apparent that the line passing ∇e_{l1} and perpendicular to **r** separates the origin from all other gradient points. This implies that the direction angles of all $\nabla e_k - \nabla e_{l1}$ ($k\neq l1$) are within a scope of π. We select the two with the smallest and largest angles, $l2$ is then chosen between them by testing that the line determined by it and ∇e_{l1} separates the other from the origin.

Starting from ($F0$, $G0$) with $l1$ and $l2$ being the equal-maximum pair, we search the curve $e_{l1}=e_{l2}$ down the decreasing direction until at some point ($F1$, $G1$) there is another $l3$, $0\leq l3<2M$-1, so that $e_{l1}(F1,G1)=e_{l2}(F1,G1)=e_{l3}(F1,G1)$, or until the search meets a side of R. We call this searching mode *equal-maximum search*.

(1b) There are two equal maxima at ($F0$, $G0$), say $e_{l1}(F0,G0)=e_{l2}(F0,G0)$. We do an equal-maximum search starting at ($F0$, $G0$) with $e_{l1}=e_{l2}$.

(1c) There is one maximum at ($F0$, $G0$), say $e_{l1}(F0,G0)$. We calculate the gradient ∇e_{l1}. Starting from ($F0$, $G0$), we search down the direction∇e_{l1} until at some point ($F1$, $G1$) there is another $l2$, $0\leq l2<2M$-1, so that $e_{l1}(F1,G1)=e_{l2}(F1,G1)$, or until the search meets a side of R. We call this searching mode *single-maximum search*.

D.2.2.2 (F0, G0) being on one side of R.

Let it be the side stating from (F_n, G_n). Denote the other end of that side as n_+, so $n_+=n+1$ if $n<N-1$, $n_+=0$ if $n=N-1$. The searching direction is constrained to a scope of π.

(2a) There are more than two equal maxima at ($F0$, $G0$). This is a very rare case. Let the maxima be $e_{l_1}(F0,G0)=e_{l_2}(F0,G0)=\ldots=e_{l_K}(F0,G0)$, $K>2$. We have shown that if ($F0$, $G0$) is not the minimal maximum, then there exists at least one pair of $l1$ and $l2$

so that down the decreasing direction $e_{l1}=e_{l2}$, e_{l1} and e_{l2} remain maximal. If any of these pairs satisfies that the decreasing direction of $e_{l1}=e_{l2}$ points inside R, than we do the equal-maximum search starting at ($F0$, $G0$), down $e_{l1}=e_{l2}$. Unlike the previous case (1a), this time we need to pick out all the ($l1$, $l2$) pairs that remain maximal down the curve $\mathrm{e}_{l1}=\mathrm{e}_{l2}$. From the proof of proposition D.3 we derive the follow corollary.

Proposition D.4: if $e_1=e_2=\ldots=e_K$ are K equal maxima at ($F0$, $G0$)$\in$R, $K>2$, and let H be the convex hull of the gradient points ∇e_1, ∇e_2, …, ∇e_K, then $l1$ and $l2$, $1\leq l1$, $l2\leq K$, is an equal-maximum pair if and only if the line passing the gradient points ∇e_{l1} and ∇e_{l2} separates H from the origin.

The proof is trivial. Proposition D.4 assures that equal-maximum pairs appear as adjacent sides of H. To find all the equal-maximum pairs, we locate the first one as discussed after proposition D.3. Then we try to extend at end ∇e_{l1} and ∇e_{l2} in exactly the same way as we find $l2$ using $l1$. If it is successful on either end, we repeat this process, until no more points can be involved. If any of these direction points inside R, we start an equal-maximum search.

However, if all these directions point out of R, then we test the two directions along side $\mathbf{r}_n$ to see if all the K maxima decrease together along either. If they do, we denote the slowest-decreasing one e_{l1} and search down that direction until there is another $l2$, so that $e_{l1}=e_{l2}$, or until the search reaches a vertex of R. We call this a *single-maximum side search*. If on both direction along $\mathbf{r}_n$ at least one of the maxima increases, then the following proposition shows that ($F0$, $G0$) is a minimal maximum.

Proposition D.5: if in both directions $\mathbf{d}_1$ and $\mathbf{d}_2$, e_1, e_2, …, e_K decrease, and $\mathbf{d}_1$ and $\mathbf{d}_2$ point to different sides of vector $\mathbf{e}$, then either on $\mathbf{e}$ or on $-\mathbf{e}$, e_1, e_2, …, e_K decrease.

Let the direction angles of $\mathbf{e}$ be 0, of $\mathbf{d}_1$ be η_1, of $\mathbf{d}_2$ be η_2, of $-\nabla \mathrm{e}_k$ be θ_k, without loss of generality, let $0<\eta_1<\pi$, $-\pi<\eta_2<0$. Apparently $\mathbf{d}_1$ cannot be directly opposite $\mathbf{d}_2$, so either $\eta_1<\eta_2+\pi$ or $\eta_1>\eta_2+\pi$. First let $\eta_1<\eta_2+\pi$. Then $\forall$ k, from $-\nabla e_k\cdot\mathbf{d}_1>0$ and $-\nabla e_k\cdot\mathbf{d}_2>0$ we know that $-\pi/2<\eta_1-\pi/2<\theta_k<\eta_2+\pi/2<\pi/2$. Therefore along $\mathbf{e}$, e_1, e_2, …, e_K decrease. Similarly, if $\eta_1>\eta_2+\pi$, we wrap the angles below 0 by $+2\pi$, so η_2 becomes

η_2'=η_2+2π <η_1+π, π<η_2'<2π. Then $\forall k$, from -$\nabla e_k\cdot$**d**$_1$>0 and -$\nabla e_k\cdot$**d**$_2$>0 we know that π/2<η_2'-π/2<θ_k'<η_1'+π/2<3π/2. Therefore along -**e**, e_1, e_2, …, e_K decrease. ■

Comment: To determine whether it is **e** or –**e** down which e_1, e_2, …, e_K decrease, we draw a line l so that **d$_1$** and **d$_2$** lies on the same side of l, then the decreasing direction is the one of ±**e** on the same side of l.

Proposition D.5 says that if there is an equal-maximum search direction not included in D_R(F0, G0), and e_1, …, e_K do not descend together down either direction along side **r**$_n$, then they don't descend together down any direction in D_R(F0, G0). According to Proposition D.1, (F0, G0) is a minimal maximum.

(2b) There are two equal maxima at (F0, G0), say, e_{l1}=e_{l2}. This can be regarded as a special case of (2a).

(2c) There is a single maximum at (F0, G0), say, e_{l1}. We check to see if -∇e_1 points inside R. If yes, we do the single-maximum search starting at (F0, G0) down. If not, we find the decreasing direction along the side **r**$_n$, and do the single-maximum side search. However, if -∇e_1 is perpendicular to side **r**$_n$ and points outside R, then (F0, G0) is the minimal maximum, according to Proposition D.1.

F.2.2.3 (F0, G0) being a vertex.

Let this vertex be n, i.e. (F0, G0)=(F_n, G_n). We denote the next vertex n_+, and the previous vertex n_-.

(3a) There are more than two maxima at (F0, G0). This is a very rare case. Let the equal maxima be e_{l1}=e_{l2}=e_{l3}…=e_{lK}, K>2. The processing under this condition is similar to that for condition (2a). We first check for all the equal-maximum search directions at (F0, G0) to see if any is in D_R(F0, G0), and do the equal-maximum search if so. If not, we examine the searching directions along **r**$_n$ or **r**$_{n-}$. If all of e_1, e_2, …, e_K descend in either direction, we do the single-maximum side search. If not, the following corollary ascertains that (F0, G0) is a minimal maximum.

Corollary of Proposition D.5: if in both directions **d$_1$** and **d**$_2$, e_1, e_2, …, e_K decrease, and the vectors **e**$_a$ and **e**$_b$ form an angle less than two right angles, so that **d**$_1$ is inside this angle and **d**$_2$ is outside, then in either direction **e**$_a$ or **e**$_b$, e_1, e_2, …, e_K decrease.

We consider whether $\mathbf{d}_1$ and $\mathbf{d}_2$ lie on the same side of $\mathbf{e}_a$. If they do, then they must be on different sides of $\mathbf{e}_b$. According to the comment after proposition D.5, down $\mathbf{e}_b$, e_1, e_2, …, e_K decrease. On the contrary, if $\mathbf{d}_1$ and $\mathbf{d}_2$ lie on different sides of $\mathbf{e}_a$, Proposition D.5 ensures that e_1, e_2, …, e_K decrease either in direction $\mathbf{e}_a$ if $\mathbf{d}_1$ and $\mathbf{d}_2$ lie on the same side of $\mathbf{e}_b$, or in the direction of -$\mathbf{e}_a$ if $\mathbf{d}_1$ and $\mathbf{d}_2$ lie on different sides of $\mathbf{e}_b$. In the latter case, we apply Proposition D.5 regarding directions -$\mathbf{e}_a$ and $\mathbf{r}_1$ and prove that e_1, e_2, …, e_K decrease in direction $\mathbf{e}_b$. ■

(3b) There are two equal maxima at (F0, G0). This is a very rare case, and can be regarded as a special case of (3a).

(3c) There is a single maximum at (F0, G0), say, e_{l1}. We check to see if $-\nabla e_1$ points inside R. If yes, we do the single-maximum search starting from (F0, G0). If not, we check the two directions $\mathbf{r}_n$ and $\mathbf{r}_{n-}$. If e_{l1} decreases in either direction, we do the single-maximum side search along the corresponding side. If not, (F0, G0) is a minimal maximum, according to the corollary above.

D.2.3 The three searching modes

Now we have finished discussing the nine conditions at the (F0, G0). There are three types of searching involved. We detail them as follows.

D.2.3.1 Equal-maximum search

Starting at (F0, G0) with $e_{l1}=e_{l2}$ being the maximum, move along the curve $e_{l1}=e_{l2}$ in the decreasing direction until at some point (F1, G1) there is another l_3 so that $e_{l3}=e_{l1}=e_{l2}$, or the search meets a side of R.

This is the only curve search of the three types, i.e. the searching route $e_{l1}=e_{l2}$ is not a straight line. $e_{l1}=e_{l2}$ may have up to 2 intersections with $e_{l1}=e_k$, $k\neq l1$, $k\neq l2$, as well as up to 2 intersections with any side of R. Apparently (F1, G1) shall be chosen as the $e_{l1}=e_{l2}=e_k$ point closest to (F0, G0) on the decreasing side, or, if it is outside R, the intersection of $e_{l1}=e_{l2}$ with a side of R between this point and (F0, G0). The intersections are solved using procedures discussed in D.2.3.4 (A).

After we get the solution of e_{l1}=e_{l2}=e_k closest to (F0, G0), let it be (F2, G2), we test if (F2, G2) is inside R. This is done by testing (F2, G2) is on the polygon side of each side $\mathbf{r}_n$ of R. If (F2, G2) lies inside R, we set (F1, G1)=(F2, G2), and start next step with starting point (F1, G1) inside R with 3 equal maxima (condition 1a). If (F2, G2) is out of side $\mathbf{r}_n$, we solve for intersections of e_{l1}=e_{l2} with $\mathbf{r}_n$ using the method in C.2.3.4 (B). As (F0, G0) and (F2, G2) are on two sides on $\mathbf{r}_n$, it is guaranteed that there exist an (F3, G3) being on *line* $\mathbf{r}_n$ and on curve e_{l1}=e_{l2} between (F0, G0) and (F2, G2). In most cases (F3, G3) is on *side* $\mathbf{r}_n$, then we set (F1, G1)=(F3, G3). However, if (F3, G3) is not on side $\mathbf{r}_n$, it must be outside either the previous or the next side, let it be $\mathbf{r}_m$. Then (F3, G3) and (F0, G0), both on curve e_{l1}=e_{l2}, are on different sides of $\mathbf{r}_m$, hence we can solve for an intersection of e_{l1}=e_{l2} with $\mathbf{r}_m$, say (F4, G4). This process can carry on until we find the first intersection of e_{l1}=e_{l2} with a side of R, let it be (F1, G1). We start the next step at (F1, G1) with two equal maxima (condition 2b).

D.2.3.2 Single-maximum search

Starting at (F0, G0) with e_{l1} being the maximum, move down a given direction (typically $-\nabla e_{l1}$), in which e_{l1} decreases, until at some point (F1, G1) there is another l2 so that $e_{l2}=e_{l1}$, or the search meets a side of R.

The meeting point of the searching path with $e_{l2}=e_{l1}$ can be solved by the method in D.2.3.4 (B). We pick the point that is closest to (F0, G0) down the decreasing side, let it be (F2, G2). We then test if (F2, G2) is inside R. If it is, we start the next step at (F2, G2) with two equal maxima (condition 1b). If not, find the intersection of the searching path with R, which is very similar to the process discussed in D.2.3.1. Let it be (F1, G1). We then start the next step at (F1, G1) with one maximum (condition 2c).

D.2.3.3 Single-maximum side search

Starting at (F0, G0) on side $\mathbf{r}_n$ of R with e_{l1} being the maximum, move along that side down the decreasing direction of e_{l1}, until at some point (F1, G1) there is another l2 so that $e_{l2}=e_{l1}$, or the search reaches one end of $\mathbf{r}_n$.

The meeting point of the $\mathbf{r}_n$ with $e_{l2} = e_{l1}$ can be solved by the method in D.2.3.4 (B). We pick the point that is closest to (*F*0, *G*0) down the decreasing side. Let it be (*F*2, *G*2). If this point is within side R, then we start the next step at (*F*2, *G*2) with two equal maxima (condition 2b); otherwise we start the next step at the vertex at the end of $\mathbf{r}_n$ in the decreasing direction (condition 3c).

D.2.3.4 Calculating the intersections

A. Solving equations $e_{l1}(F, G) = e_{l2}(F, G) = e_{l3}(F, G)$.

We know that e_{l1}(F, G) is in the form of $\dfrac{m_l\sqrt{F + k_l G} - \hat{f}^{m_l}}{\Delta^{m_l}}$ or $\dfrac{\hat{f}^{m_l} - m_l\sqrt{F + k_l G}}{\Delta^{m_l}}$.

We rewrite them in a standard format:

$$e_1 = m_1'\sqrt{F + k_1 G} - f_1',\ e_2 = m_2'\sqrt{F + k_2 G} - f_2',\ e_3 = m_3'\sqrt{F + k_3 G} - f_3'. \quad \text{(D. 9a)}$$

where

$$m_1' = \pm\frac{m_{l1}}{\Delta^{m_{l1}}},\ f_1' = \pm\frac{\hat{f}^{m_{l1}}}{\Delta^{m_{l1}}},\ m_2' = \pm\frac{m_{l2}}{\Delta^{m_{l2}}},\ f_2' = \pm\frac{\hat{f}^{m_{l2}}}{\Delta^{m_{l2}}},$$

$$m_3' = \pm\frac{m_{l3}}{\Delta^{m_{l3}}},\ f_3' = \pm\frac{\hat{f}^{m_{l3}}}{\Delta^{m_{l3}}}. \quad \text{(D. 9b)}$$

($\pm$: + if $m_l\sqrt{F + k_l G} - \hat{f}^{m_l}$ >0, - if $m_l\sqrt{F + k_l G} - \hat{f}^{m_l}$ <0.)

So that the equations are written as

$$m_1'\sqrt{F + k_1 G} - f_1' = m_2'\sqrt{F + k_2 G} - f_2',\ m_1'\sqrt{F + k_1 G} - f_1' = m_3'\sqrt{F + k_3 G} - f_3'. \text{(D. 9c)}$$

Substitute with

$$x = \sqrt{F + k_1 G} \quad \text{(D. 10)}$$

we get

$$m_1'x - f_1' = m_2'\sqrt{x^2 + h_{21}G} - f_2',\ m_1'x - f_1' = m_3'\sqrt{x^2 + h_{31}G} - f_3'. \quad \text{(D. 11a)}$$

where

$$h_{21} = k_2 - k_1 = m_2^2 - m_1^2,\ h_{31} = k_3 - k_1 = m_3^2 - m_1^2. \quad \text{(D. 11b)}$$

Then

$$m_1'x + f_2' - f_1' = m_2'\sqrt{x^2 + h_{21}G}\ ,\ m_1'x + f_3' - f_1' = m_3'\sqrt{x^2 + h_{31}G}\ . \qquad \text{(D. 12)}$$

Then

$$(m_1'^2 - m_2'^2)x^2 + 2m_1'(f_2' - f_1')x + (f_2' - f_1')^2 = m_2'^2 h_{21}G\,,$$

$$(m_1'^2 - m_3'^2)x^2 + 2m_1'(f_3' - f_1')x + (f_3' - f_1')^2 = m_3'^2 h_{31}G\,. \qquad \text{(D. 13a)}$$

Let

$$h_{12}' = m_1'^2 - m_2'^2\,,\ h_{13}' = m_1'^2 - m_3'^2\,, \qquad \text{(D. 14a)}$$

eliminating x^2 we get:

$$\begin{aligned} &2m_1'\left(h_{13}'(f_2' - f_1') - h_{12}'(f_3' - f_1')\right)x + h_{13}'(f_2' - f_1')^2 - h_{12}'(f_3' - f_1')^2 \\ &= \left(m_2'^2 h_{13}' h_{21} - m_3'^2 h_{12}' h_{31}\right)G \end{aligned}\ . \qquad \text{(D. 14b)}$$

That is

$$G=ax+b, \qquad \text{(D. 15a)}$$

where

$$a = \frac{2m_1'\left(h_{13}'(f_2' - f_1') - h_{12}'(f_3' - f_1')\right)}{m_2'^2 h_{13}' h_{21} - m_3'^2 h_{12}' h_{31}}\,,\ b = \frac{h_{13}'(f_2' - f_1')^2 - h_{12}'(f_3' - f_1')^2}{m_2'^2 h_{13}' h_{21} - m_3'^2 h_{12}' h_{31}}\,. \qquad \text{(D. 15b)}$$

We then have a simple quadratic equation

$$h_{12}'x^2 + \left(2m_1'(f_2' - f_1') - m_2'^2 h_{21}a\right)x + (f_2' - f_1')^2 - m_2'^2 h_{21}b = 0\,. \qquad \text{(D. 16)}$$

After solving x we calculate $G = ax+b$.

However, if it happens that $m_2'^2 h_{13}' h_{21} - m_3'^2 h_{12}' h_{31} = 0$ we cannot derive (D. 15b). In this case we derive from (D.14b)

$$x = \frac{h_{12}'(f_3' - f_1')^2 - h_{13}'(f_2' - f_1')^2}{2m_1'\left(h_{13}'(f_2' - f_1') - h_{12}'(f_3' - f_1')\right)}\,, \qquad \text{(D. 17a)}$$

then calculate

$$G = \frac{h'_{12}x^2 + 2m'_1(f'_2 - f'_1)x + (f'_2 - f'_1)^2}{m'^2_2 h_{21}} \text{ if } h_{21} \neq 0, \qquad \text{(D. 17b)}$$

or

$$G = \frac{h'_{13}x^2 + 2m'_1(f'_3 - f'_1)x + (f'_3 - f'_1)^2}{m'^2_3 h_{31}} \text{ if } h_{31} \neq 0. \qquad \text{(D. 17c)}$$

After calculating x and G, *we* calculate F by $F= x^2-k_1G$.

B. solving equation $e_{l1}(F_{-1}+\lambda\delta F, G_{-1}+\lambda\delta G) = e_{l2}(F_{-1}+\lambda\delta F, G_{-1}+\lambda\delta G)$.

We know that $e_{ll}(F, G)$ is in the form of $\frac{m_l\sqrt{F+k_lG} - \hat{f}^{m_l}}{\Delta^{m_l}}$ or $\frac{\hat{f}^{m_l} - m_l\sqrt{F+k_lG}}{\Delta^{m_l}}$.

We rewrite in a standard format:

$$e_{l1} = m'_1\sqrt{F+k_1G} - f'_1,\ e_{l2} = m'_2\sqrt{F+k_2G} - f'_2, \qquad \text{(D. 18a)}$$

where

$$m'_1 = \pm\frac{m_{l1}}{\Delta^{m_{l1}}},\ f'_1 = \pm\frac{\hat{f}^{m_{l1}}}{\Delta^{m_{l1}}},\ m'_2 = \pm\frac{m_{l2}}{\Delta^{m_{l2}}},\ f'_2 = \pm\frac{\hat{f}^{m_{l2}}}{\Delta^{m_{l2}}}. \qquad \text{(D. 18b)}$$

$(\pm$: + if $m_l\sqrt{F+k_lG} - \hat{f}^{m_l} > 0$, - if $m_l\sqrt{F+k_lG} - \hat{f}^{m_l} < 0)$.

Then our equation becomes

$$\begin{aligned} & m'_1\sqrt{(\delta F + k_1\delta G)\lambda + (F_{-1} + k_1G_{-1})} - f'_1 \\ &= m'_2\sqrt{(\delta F + k_2\delta G)\lambda + (F_{-1} + k_2G_{-1})} - f'_2 \end{aligned} \qquad \text{(D. 18c)}$$

Substitute

$$x = \sqrt{(\delta F + k_1\delta G)\lambda + (F_{-1} + k_1G_{-1})}, \qquad \text{(D. 19a)}$$

then

$$x^2 = (\delta F + k_1\delta G)\lambda + (F_{-1} + k_1G_{-1}) \qquad \text{(D. 19b)}$$

and

$$
\begin{aligned}
&(\delta F + k_2\delta G)\lambda + (F_{-1} + k_2G_{-1}) \\
&= \frac{\delta F_{-1} + k_2\delta G}{\delta F_{-1} + k_1\delta G}x^2 + (F_{-1} + k_2G_{-1}) - \frac{\delta F + k_2\delta G}{\delta F + k_1\delta G}(F_{-1} + k_1G_{-1})\,. \qquad \text{(D. 19c)} \\
&= \frac{(\delta F + k_2\delta G)x^2 + (k_1 - k_2)(F_{-1}\delta G - G_{-1}\delta F)}{\delta F + k_1\delta G}
\end{aligned}
$$

The equation becomes

$$
m_1'x - f_1' = m_2'\sqrt{\frac{(\delta F + k_2\delta G)x^2 + (k_1 - k_2)(F_{-1}\delta G - G_{-1}\delta F)}{\delta F + k_1\delta G}} - f_2'\,, \qquad \text{(D. 20)}
$$

This is then written as a simple quadratic equation regarding *x*:

$$
\begin{aligned}
&\left(m_1'^2 - \frac{\delta F + k_2\delta G}{\delta F + k_1\delta G}m_2'^2\right)x^2 + 2m_1'(f_2' - f_1')x \\
&\qquad + \left((f_2' - f_1')^2 - \frac{(k_1 - k_2)(F_{-1}\delta G - G_{-1}\delta F)}{\delta F + k_1\delta G}m_2'^2\right) = 0
\end{aligned}
\,. \qquad \text{(D. 21)}
$$

After solving *x* we can calculate

$$
\lambda = \frac{x^2 - (F_{-1} + k_1G_{-1})}{\delta F + k_1\delta G}\,,\ F = F_{-1} + \lambda\delta F\,,\ G = G_{-1} + \lambda\delta G\,. \qquad \text{(D. 22)}
$$

When we search down the reverse gradient direction, let the gradient direction be of e_{11} be ±(1, k_1), then $\delta F = \mp 1$, $\delta G = \mp k_1$, and the equation of *x* is

$$
\left(m_1'^2 - \frac{1 + k_1k_2}{1 + k_1^2}m_2'^2\right)x^2 + 2m_1'(f_2' - f_1')x + \left((f_2' - f_1')^2 - \frac{(k_1 - k_2)(k_1F_{-1} - G_{-1})}{1 + k_1^2}m_2'^2\right) = 0\,.
$$

(D. 23a)

After solving *x* we calculate

$$
\lambda = \mp\frac{x^2 - (F_{-1} + k_1G_{-1})}{1 + k_1^2}\,,\ F = F_{-1} \mp \lambda\,,\ G = G_{-1} \mp k_1\lambda\,. \qquad \text{(D. 23b)}
$$

Appendix E

Calculations in the re-estimation of sinusoids

E.1 De-averaging amplitudes

An instantaneous amplitude $\hat{a}$ estimated using (3.34a) can be written as the weighted average of the instantaneous amplitude within the frame:

$$\hat{a} = \frac{\sum_{n=0}^{N-1} w_n^2 a_n}{\sum_{n=0}^{N-1} w_n^2} \tag{3. 34b}$$

where w is the window function used in (3.34a).

We formulate the de-averaging of (3.34b) as follows. Given the instantaneous amplitude estimates $\hat{a}_0$, $\hat{a}_h$, …, $\hat{a}_{Lh}$, find a_0, …, a_{Lh}, so that by interpolating a_0, …, a_{Lh} as $\{a_n\}_{n=-N/2, \ldots, Lh+N/2}$, we have

$$\hat{a}_{lh} = \frac{\sum_{n=0}^{N-1} w_n^2 a_{lh-N/2+n}}{\sum_{n=0}^{N-1} w_n^2}, \; l=0, 1, \ldots, L. \tag{E. 1}$$

In this de-averaging problem we have L+1 variables and L+1 equations. Obviously the de-averaging process significantly depends on the interpolation method. We assume that the interpolated amplitude is linear time-invariant regarding a_0, …, a_{Lh}, i.e.

$$a_n = \sum_l a_{lh} g_n^l \, . \tag{E. 2a}$$

g is known as the *interpolation kernel*, and

$$g_{kh}^l = \delta(k-l) \, , \quad g_n^{l+m} = g_{n-mh}^l \tag{E. 2b}$$

From the second equation we immediately have

$$a_n = \sum_{l=0}^{L} a_{lh} g_{n-lh}^{0} \tag{E. 2c}$$

This is a convolution with the impulse response *g*. In the context of slow amplitude variation, *g* is always low-pass.

As an example we look at the quadratic interpolation with overlap-add. Given three consecutive amplitude estimates $a_{(l-1)h}$, a_{lh}, $a_{(l+1)h}$, we can use a parabolic function to interpolate between $(l\text{-}1)h$ and $(l\text{-}1)h$ as

$$a^{l}(lh+t) = a_{(l-1)h}\frac{t(t-h)}{2h^2} - a_{lh}\frac{(t-h)(t+h)}{h^2} + a_{(l+1)h}\frac{t(t+h)}{2h^2} \tag{E. 3}$$

An interpolation of the amplitude track can be implemented by overlap-adding these parabolic amplitudes:

$$a(t) = \sum_{l=0}^{L} a^{l}(t)v(t-lh) \tag{E. 4}$$

where the overlap-add window *v* satisfies

$$v(t) = 0, \forall t \ge h \tag{E. 5a}$$

$$v(t) = v(-t) \tag{E. 5b}$$

$$v'(h) = v''(h) = 0 \tag{E. 5c}$$

$$v(t) + v(t+h) = 1, \quad \forall -h \le t \le 0 \tag{E. 5d}$$

It can be shown that

$$\sum_{l=-\infty}^{\infty} v(t-lh) = 1 \tag{E. 5e}$$

The interpolated amplitude track (E.4) has continuous 1st- and 2nd-order derivatives. Combining (E.3) and (E.4) we get

$$\begin{aligned} a_{kh+n} &= a_{(k-1)h}\frac{n(n-h)v_n}{2h^2} + a_{kh}\left(\frac{(n-h)\left(v_{n-h}(n-2h) - v_n 2(n+h)\right)}{2h^2}\right) \\ &+ a_{(k+1)h}\left(\frac{n\left(v_n(n+h) - v_{n-h}2(n-2h)\right)}{2h^2}\right) + a_{(k+2)h}\frac{(n-h)nv_{n-h}}{2h^2}, \quad 0 \le n < h \end{aligned} \tag{E. 6}$$

Compare (E.2a) and (E.6) we get

$$g_n^l = \begin{cases} \dfrac{(n \backslash h)(n \backslash h - h)v_{n\backslash h}}{2h^2}, & l = \left\lfloor \dfrac{n}{h} \right\rfloor - 1 \\[2ex] \dfrac{(n \backslash h - h)\big((n \backslash h - 2h)v_{n\backslash h-h} - 2(n \backslash h + h)v_{n\backslash h}\big)}{2h^2}, & l = \left\lfloor \dfrac{n}{h} \right\rfloor \\[2ex] \dfrac{(n \backslash h)\big((n \backslash h + h)v_{n\backslash h} - 2(n \backslash h - 2h)v_{n\backslash h-h}\big)}{2h^2}, & l = \left\lfloor \dfrac{n}{h} \right\rfloor + 1 \\[2ex] \dfrac{(n \backslash h - h)(n \backslash h)v_{n\backslash h-h}}{2h^2}, & l = \left\lfloor \dfrac{n}{h} \right\rfloor + 2 \\[2ex] 0, & otherwise \end{cases} \tag{E. 7a}$$

where “\” is the modulo operator and “$\lfloor\ \rfloor$” is the integer floor operator. A more practical computation of g is

$$g_{lh+n}^k = \begin{cases} \dfrac{n(n-h)v_n}{2h^2}, & k = l-1 \\[2ex] \dfrac{(n-h)\big((n-2h)v_{n-h} - 2(n+h)v_n\big)}{2h^2}, & k = l \\[2ex] \dfrac{n\big((n+h)v_n - 2(n-2h)v_{n-h}\big)}{2h^2}, & k = l+1 \\[2ex] \dfrac{(n-h)(n)v_{n-h}}{2h^2}, & k = l+2 \\[2ex] 0, & otherwise \end{cases} \tag{E. 7b}$$

Combining (E.1) and (E.2a) we get

$$\hat{a}_{lh} = \sum_{k=0}^{L} \left(\frac{\sum_{n=0}^{N-1} w_n^2 g_{lh-N/2+n}^k}{\sum_{n=0}^{N-1} w_n^2} \right) a_{kh}, \ l\text{=0, 1, …, L.} \tag{E. 8}$$

This is a multi-diagonal linear system of order L+1. By solving this system we get the de-averaged amplitude.

E.2 De-averaging frequencies

An instantaneous frequency $\hat{f}$ estimated using the LSE method can be written as the weighted average of the instantaneous frequency within the frame (3.9c). For de-averaging we use (3.33) and rearranging the summing indices as:

$$\hat{f}_{lh} = \frac{\sum_{n=1}^{N-1}\sum_{m=0}^{n-1} a_{mn}^{l} \int_{m}^{n} f(t)dt}{\sum_{n=1}^{N-1}\sum_{m=0}^{n-1} a_{mn}^{l}(n-m)}, \ l=0,\ 1,\ \ldots,\ L, \qquad \text{(E. 9a)}$$

where

$$a_{mn}^{l} = (n-m)w_m^2 w_n^2 a_{lh-N/2+n} a_{lh-N/2+m} \operatorname{sinc} 2\left(\int_m^n f(lh-N/2+t)dt - (n-m)\hat{f}_{lh}\right) \text{(E. 9b)}$$

w is the window function used in the LSE estimation. Unlike the amplitude case, for frequency the averaging weights a_{mn}^{l} depend both on the estimate $\hat{f}$ itself and on the frequency track $f(t)$. In the re-estimation process the dependency on $\hat{f}$ is not a problem since it is already known. However, the dependency on $f(t)$ makes the de-averaging problem highly non-linear. In the iterative framework we may use the frequency from the last iteration, i.e. an $\hat{f}(t)$ interpolated from $\{\hat{f}_{lh}\}_{l=0,\cdots,L}$, to calculate the averaging weights, so that the weights have no dependency on $f(t)$.

Like the amplitude interpolation, we assume that the frequency interpolation is linear regarding the samples, i.e.

$$f(t) = \sum_{l} f_{lh} g^{l}(t), \ \hat{f}(t) = \sum_{l} \hat{f}_{lh} g^{l}(t) \qquad \text{(E. 10)}$$

where g is the interpolation kernel. Then we can calculate

$$a_{mn}^{l} = (n-m)w_m^2 w_n^2 a_{lh-N/2+n} a_{lh-N/2+m} \operatorname{sinc} 2\left(\sum_{i} \hat{f}_{ih} \int_m^n g^{i}(lh-N/2+t)dt - (n-m)\hat{f}_{lh}\right)$$

(E. 11)

Combining (E. 9a) and (E. 10) we get

$$\hat{f}_{lh} = \sum_{k=0}^{l} \left(\frac{\sum_{n=1}^{N-1}\sum_{m=0}^{n-1} a_{mn}^{l} \int_{m}^{n} g^{k}(t)dt}{\sum_{n=1}^{N-1}\sum_{m=0}^{n-1} a_{mn}^{l}(n-m)} \right) f_{kh}\text{ , } l=0, 1, \ldots, L, \qquad \text{(E. 12)}$$

This is again a multi-diagonal linear system of order L+1. By solving this system we get the de-averaged frequency.

E.3 De-variation method

As discussed in Chapter 3, large parameter estimation errors occur as the result of large parameter dynamics. The de-averaging method is proposed by observing that the frequency estimate on a time-varying sinusoid is a weighted average of local instantaneous frequencies (see (3.33)), and the amplitude estimated from (3.34a) is a weighted average of local instantaneous amplitudes (see (3.34b)). When a rough sinusoid track is known, using these equations we know what the weights are like, and therefore are able to invert the averaging process.

The de-variation method, alternatively, tries to remove the variations from the parameters using the rough sinusoid track. The idea is given in Figure E.1. The solid line is the true signal track that is not directly observable, and the dashed line is the estimated track. If we subtract the dashed track from the solid one, we get the red track, which has a much slower variation and much easier to estimate accurately. After we have estimated this “difference track”, it can be simply added back to the original estimate, i.e. the dashed track, as a re-estimation. Like the de-averaging method, this de-variation method can be performed iteratively.

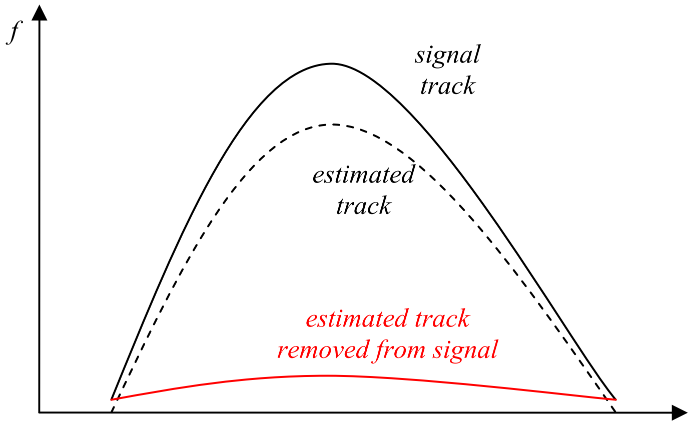


Figure E. 1 De-variation method

The de-variation method proceeds as follows. Let F^i stand for a set of frequency estimates, A^i stand for a set of amplitude estimates, $P^i=\{F^i, A^i\}$. $F^0=\{\hat{f}_0, \hat{f}_1, \dots, \hat{f}_L\}$. For i=1, 2, …, do 1~5, until Δ is below some threshold, or i is above a maximal number of iterations:

1) interpolate the frequency estimates $F^{i\text{-}1}$ as $f^{i-1}(t)$, and the amplitude estimates $\mathrm{A}^{i\text{-}1}$ as $a^{i\text{-}1}(t)$;

2) for l=0, 1, …, L, do 3 and 4;

3) let x, f, and a be the signal, interpolated frequency tracks, and interpolated amplitude track, of the l^{th} frame, all shifted to centre at 0, calculate

$$y_n = \frac{a_0}{a_n} x_n e^{-j2\pi\int_0^n (f(t)-f(0))d(t)} \tag{E. 13}$$

(y is the de-variationed version of x, where the frequency variation is removed by the multiplication with a phase term, and the amplitude variation is removed by the division using the interpolated amplitude track. In particular, if the interpolated tracks are accurate and x is noise-free, then y is a constant.)

4) estimate the frequency, amplitude and phase angle of y;

5) let P^i be the collection of the new estimates evaluated in the above loop, calculate the distance Δ between $P^{i\text{-}1}$ and P^i.

When the iteration finishes the final P^i is returned as the re-estimated result. The de-variation method is more general than the de-averaging method as there is no constraint on what estimator to use. If the LSE method is used, the method can be implemented in a stable way by keeping watch on the square error, so that the error never grows from one iteration to the next.

Appendix F

Spectral domain resynthesis

The direct synthesis method, when applied to the interval n_l~n_{l+1}, requires calculating Δn_l points for each partial. When the number of partials is large, the computation cost becomes heavy. However, in frequency domain a sinusoid appears within a very small band, and it is possible to approximate its spectrum with only a few operations. The spectral resynthesis of a harmonic sinusoid builds the short-time Fourier transform of the combination of sinusoids, then use inverse DFT to get time-domain resynthesis. To connect the rebuilt signals smoothly at the frame boundaries, an overlap-add method is used. [MQ86] suggests that the overlap rate should be kept high for good results. In our system we use an overlap rate of 50%, but allow multiple frame widths. This multi-resolution overlap-add synthesis may accompany the multi-resolution re-estimation method detailed (§3.4), or stand alone using single-resolution estimates.

F.1 Overlap-add

In frame-based synthesis the overlap-add (OLA) method is often used for connecting frames to eliminate abrupt discontinuity [AR77]. In spectral synthesis we use a 1/2 overlap rate, which also requires 1/2 overlap on the analyzer side. That is, let $2h$ be the window size for the DFT and let it be constant, then the hop size between adjacent windows is set to h. The measurement points are chosen at h, $2h$, …, $l{\cdot}h$, …, with the l^{th} frame centred at $l{\cdot}h$, spanning the duration of $2h$ from $(l\text{-}1)h$ to $(l\text{+}1)h$. Let the rebuild signal of the l^{th} frame be $\tilde{x}_l = [\tilde{x}_{l,0}, \tilde{x}_{l,1}, \cdots, \tilde{x}_{l,2h}]$, and let w be a low-pass synthesis window function of size $2h$, spanning the duration from 0 to $2h$, then in the OLA method $\tilde{x}$ is rebuilt as

$$\tilde{x}_n = \sum_l \tilde{x}_{l,n-(l-1)h} w_{n-(l-1)h} \tag{F. 1}$$

That is, we weight the resynthesized frame using window function *w*, align each frame to its expected position, then sum them up. *w* is chosen so that 1) it is non-negative and symmetric, 2) it has a maximum at the central point *h*, 3) it fades away at both ends, and 4) they overlap-add to identity, i.e.

$$\sum_{l} w_{n-(l-1)h} = 1 \tag{F. 2}$$

Examples of window functions that meet all the conditions include the Hann window and triangular window.

The OLA processing is *h*-shift invariant, that is, shifting the time axis by a multiple of *h* does not disturb the resynthesis result. However, only when *h*=1 is it shift invariant, otherwise it introduces a modulation artefact with period *h*. An overlap-add window that satisfies (F.2) leads to the constant OLA, which eliminates amplitude modulation due to the shift variance on stationary sinusoids.

Since the overlap-add window is low-pass, it typically has faster decay than the sinc function, which characterize the DFT of a sinusoid. Since the computation cost of synthesizing the spectrum is roughly proportional to the number of bins in the resynthesized spectrum, by synthesizing the spectrum of a *windowed* sinusoid we are able to save computation by using fewer bins, as when the spectrum decays faster the energy is more concentrated.

F.2 Quasi-stationary partials

When a partial varies very slowly within the considered frame, we approximate its spectrum with that of a constant sinusoid, that is

$$\widetilde{X}_k = \begin{cases} \hat{a}e^{j\hat{\varphi}} \cdot W(k/N - \hat{f}), & |k - N\hat{f}| \leq \mathrm{B} \\ 0, |k - N\hat{f}| > \mathrm{B} \end{cases} \tag{F. 3a}$$

Constant B indicates how many bins are in the resynthesis. All bins that are more than B bins from the frequency estimate are discarded.

Let $\{\hat{a}_l^m, \hat{f}_l^m, \hat{\varphi}_l^m\}$ be the parameter set estimated for the *m*th partial of frame *l*, then we synthesize its spectrum using (F.3a)

$$(\tilde{X}_l^m)_k = \begin{cases} \hat{a}_l^m e^{j\hat{\varphi}_l^m} \cdot W(k/N - \hat{f}_l^m), \quad |k - N\hat{f}_l^m| \leq \mathrm{B} \\ \\ 0, |k - N\hat{f}_l^m| > \mathrm{B} \end{cases} \qquad \text{(F. 3b)}$$

The summary windowed spectrum $\tilde{X}_l$ is then calculated as

$$\tilde{X}_{l,k} = \sum_m (\tilde{X}_l^m)_k \qquad \text{(F. 3c)}$$

F.3 Fast-varying partials

In the overlap-add method each frame is independently synthesized as a stationary sinusoid without frequency or amplitude variation. While the overlap-add process interpolate between two frames in time domain, there is no explicit interpolation on amplitude or frequency. When the frequency dynamics is low, the resynthesized windowed spectra of individual frames overlap in the time-frequency plane (Figure F.1, (a)). When they are summed up the spectrum covers the complete range of the instantaneous frequency, and gives a smooth adaptation between parameter sets. However, when the frequency dynamics is high, e.g. the frequency jump between frames is larger than with which the windowed spectra may overlap (Figure F.1, (b)), then the resynthesized spectrum becomes discontinuous in the time-frequency plane, leaving large areas in the time-frequency plane, where the sinusoid pass by, empty, which is far from the typical behaviour of a time-varying sinusoid.

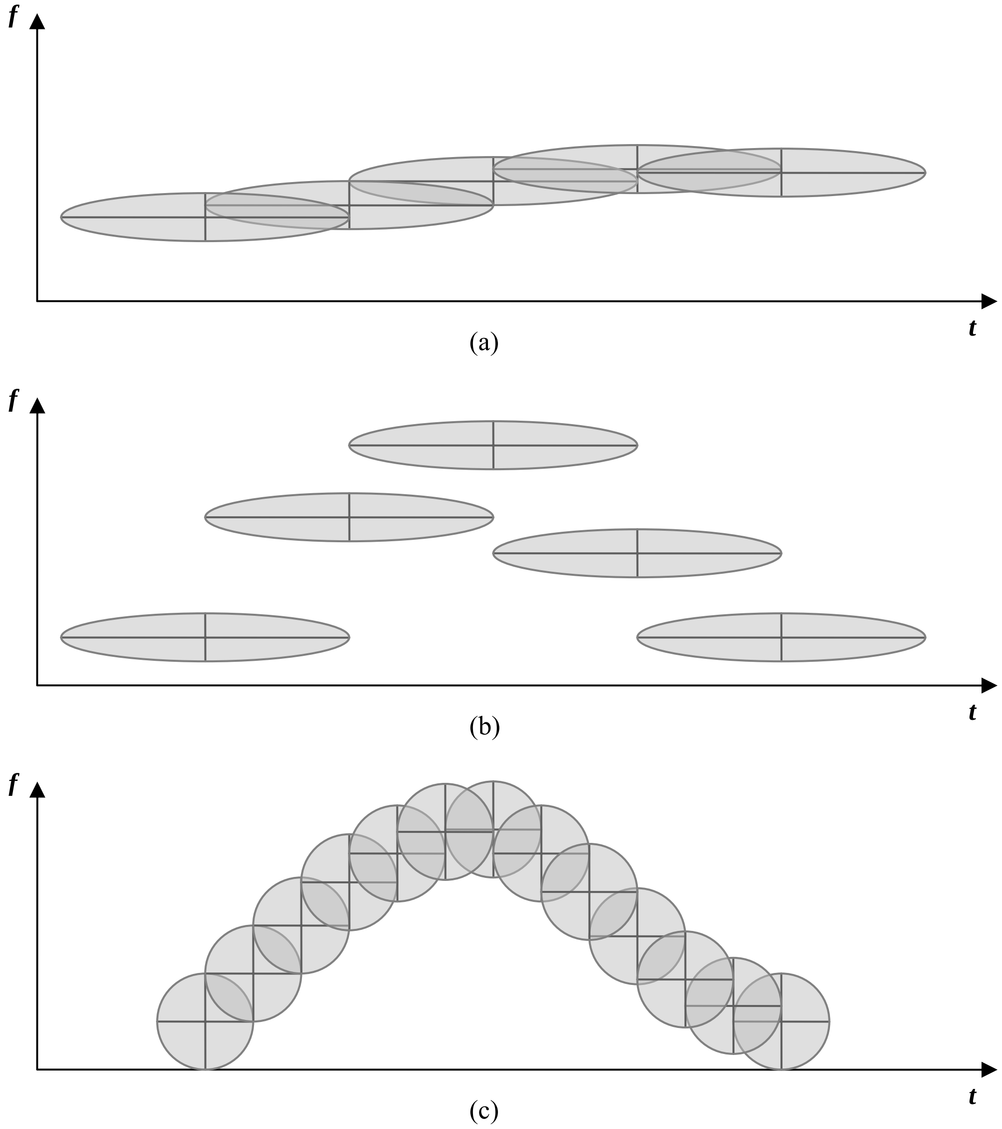


Figure F. 1 Synthesizing sinusoid particles within a partial

In Appendix A.3.2 we have divided a frame of time-varying sinusoid into several overlapping segments to estimate the spectrum. By giving each segment its own frequency centre, we are able to synthesize detailed frequency (and amplitude too) variation within a frame. Figure F.1 (c) shows this idea. Compared to (b), it uses a 1/3 window length (therefore 3 times bin width), and 3 times the number of frames. On one hand, by increasing the number of frames in the same duration, we cut down the frequency jump between adjacent frames. On the other, by reducing the window length, we increase the frequency coverage of each atom. This combination of changes makes the individual spectra meet together seamlessly in Figure F.1 (c), so that the summary spectrogram covers the complete frequency range in a way a time-varying sinusoid does.

In practice all the window widths are powers of 2. A frame of size $2h$ is always aligned to multiples of h. We call the centre of a DFT frame used in the synthesizer a *synthesis point*. Each synthesis point is associated with a window size. If the multiresolution re-estimation in §3.4 has been used, then we can use all the final measurement points as synthesize points, along their windows sizes. From each measurement point a sinusoid atom is constructed. Finally the harmonic sinusoid is constructed by summing up all the atoms.

However, even if we only have a single-resolution sinusoid track, it is also possible to do a multi-resolution synthesis. To do this we first find the synthesis points along with there window sizes, associate with each point the sinusoidal parameters obtained from interpolating between the measurement points. When this is done we can calculate the atoms, and finally, the harmonic sinusoid. The synthesis points are found by comparing the frequency jump between adjacent atoms and the bandwidths of them to see if the jump is covered within the bandwidths. We define the single-sided bandwidth of a sinusoid atom as $1/N$, where N is the window width for synthesizing the atom. For the Hann window this is the 6dB bandwidth. To select the synthesis points, we start from the measurement points, with their original window sizes, and do the following.

```
Let there be $L$+1 points in the list, located at $n_0$, …, $n_L$, with frequency estimates
    $\hat{f}_0$, …, $\hat{f}_L$, frame width $N_0$, …, $N_L$, and single-sided frequency spans $b_0$, …,
    $b_L$;
1. for $l$=0, 1, …, $L$-1, do 2~3;
    2. if $b_l+b_{l+1}<|\hat{f}_l-\hat{f}_{l+1}|$, do 3;
    3. insert a new point at $0.5\cdot(n_l+n_{l+1})$, let its frame width be $n_{l+1}-n_l$, single-sided
       frequency span be $1/(n_{l+1}-n_l)$, and get the frequency estimate at this point;
4. if no point has been inserted in the loop, then terminate the process, as all the
   synthesis points are already found;
```

```
5. for $l$=1, 3, …, $L$-1, do 6;
    6. if new points have been inserted at $0.5\cdot(n_l+n_{l+1})$ and $0.5\cdot(n_l+n_{l-1})$, then $N_l \leftarrow 0.5N_l$, $b_l \leftarrow 2b_l$;
7. let $L$, $n_l$, $\hat{f}_l$, $N_l$ and $b_l$ be redefined for the new list with inserted points, go to step 1.
```

The main point of step 6 is to smooth the sequence of window sizes. Intuitively, if there is one point inserted on each side of n_l, it implies that the frequency variation near n_l is faster than the window size N_l can catch up with, then it is natural that N_l be reduced.

To construct a signal using multiple window widths, we synthesize a spectrogram for each window width, rebuild a time-domain signal from this spectrogram using the overlap-add method, and finally sum them up. However, the constant sum criterion for the resynthesis windows is not satisfied where the window width switches between frames. This is shown in Figure F.2. In the figure the solid horizontal line represents the time axis and the triangles above this line represent synthesis windows. The dashed line marks the sum of the window functions. In (a) and (b) only one of the two window widths is used, and the windows sum up to identity. (c) is derived by inserting synthesis points between the measure points of (b), marked by the arrows. Consequently, a different window size is chosen for these points and some of the original points in (b), which results in a non-constant sum of windows. One way to preserve the constant-sum property is shown in (d). When two adjacent windows have different window widths, we shorten the longer one on the half that meets the shorter one, so that it fades off when the shorter window reaches the maximum. To implement this nonsymmetric window, one simply treat it as a full-length window with leading or trailing zeros, and calculate its DTFT $W(f)$, then use it in (F.3a) along with other frames of the same width.

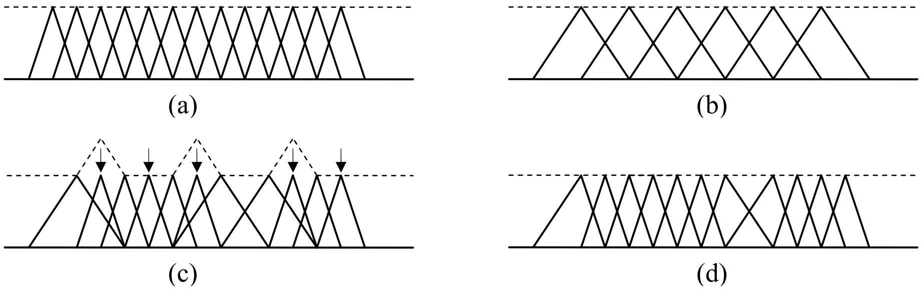


Figure F. 2 Spectral synthesis using multiple frame widths