# Adaptive Signal Models: Theory, Algorithms, and Audio Applications

by

Michael Mark Goodwin

S.B. (Massachusetts Institute of Technology) 1992
S.M. (Massachusetts Institute of Technology) 1992

A dissertation submitted in partial satisfaction of the
requirements for the degree of
Doctor of Philosophy

in

Engineering—Electrical Engineering and Computer Science

in the

GRADUATE DIVISION
of the
UNIVERSITY of CALIFORNIA, BERKELEY

Committee in charge:

Professor Edward A. Lee, Chair
Professor Martin Vetterli
Professor David Wessel

Fall 1997

The dissertation of Michael Mark Goodwin is approved:

Chair Date

Date

Date

University of California, Berkeley

Fall 1997

**Adaptive Signal Models: Theory, Algorithms, and Audio Applications**

## Abstract

Adaptive Signal Models: Theory, Algorithms, and Audio Applications

by

Michael Mark Goodwin
Doctor of Philosophy in Engineering—Electrical Engineering and Computer Science

University of California, Berkeley

Professor Edward A. Lee, Chair

Mathematical models of natural signals have long been of interest in the scientific community. A primary example is the Fourier model, which was introduced to explain the properties of blackbody radiation and has since found countless applications. In this thesis, a variety of parametric models that are tailored for representing audio signals are discussed. These modeling approaches provide compact representations that are useful for signal analysis, compression, enhancement, and modification; compaction is achieved in a given model by constructing the model in a signal-adaptive fashion.

The opening chapter of this thesis provides a review of background material related to audio signal modeling as well as an overview of current trends. Basis expansions and their shortcomings are discussed; these shortcomings motivate the use of overcomplete expansions, which can achieve improved compaction. Methods based on overcompleteness, *e.g.* best bases, adaptive wavelet packets, oversampled filter banks, and generalized time-frequency decompositions, have been receiving increased attention in the literature.

The first signal representation discussed in detail in this thesis is the sinusoidal model, which has proven useful for speech coding and music analysis-synthesis. The model is developed as a parametric extension of the short-time Fourier transform (STFT); parametrization of the STFT in terms of sinusoidal partials leads to improved compaction for evolving signals and enables a wide range of meaningful modifications. Analysis methods for the sinusoidal model are explored, and time-domain and frequency-domain synthesis techniques are considered.

In its standard form, the sinusoidal model has some difficulties representing nonstationary signals. For instance, a pre-echo artifact is introduced in the reconstruction of signal onsets. Such difficulties can be overcome by carrying out the sinusoidal model in a multiresolution framework. Two multiresolution approaches based respectively on filter banks and adaptive time segmentation are presented. A dynamic program for deriving pseudo-optimal signal-adaptive segmentations is discussed; it is shown to substantially mitigate pre-echo distortion.

In parametric methods such as the sinusoidal model, perfect reconstruction is generally not achieved in the analysis-synthesis process; there is a nonzero difference between the original and the inexact reconstruction. For high-quality synthesis, it is important to model this residual and incorporate it in the signal reconstruction to account for salient features such as breath noise in a flute sound. A method for parameterizing the sinusoidal model residual based on a perceptually motivated filter bank is considered; analysis and synthesis techniques for this residual model are given.

For pseudo-periodic signals, compaction can be achieved by incorporating the pitch in the signal model. It is shown that both the sinusoidal model and the wavelet transform can be improved by pitch-synchronous operation when the original signal is pseudo-periodic. Furthermore, approaches for representing dynamic signals having both periodic and aperiodic regions are discussed.

Both the sinusoidal model and the various pitch-synchronous methods can be interpreted as signal-adaptive expansions whose components are time-frequency atoms constructed according to parameters extracted from the signal by an analysis process. An alternative approach to deriving a compact parametric atomic decomposition is to choose the atoms in a signal-adaptive fashion from an overcomplete dictionary of parametric time-frequency atoms. Such overcomplete expansions can be arrived at using the matching pursuit algorithm. Typically, the time-frequency dictionaries used in matching pursuit consist of Gabor atoms based on a symmetric prototype window. Such symmetric atoms, however, are not well-suited for representing transient behavior, so alternative dictionaries are considered, namely dictionaries of damped sinusoids as well as dictionaries of general asymmetric atoms constructed using underlying causal and anticausal damped sinusoids. It is shown that the matching pursuit computation for either type of atom can be carried out with low-cost recursive filter banks.

In the closing chapter, the key points of the thesis are summarized. The conclusion also discusses extensions to audio coding and provides suggestions for further work related to overcomplete representations.

Professor Edward A. Lee
Dissertation Committee Chair

# Contents

# List of Figures

## Acknowledgments

The diligent reader will notice upon further inspection that the *we*-based approach to writing is not used in this thesis; for example, at no point do *we* see some result or do *we* reach some conclusion. This stylistic preference is a bit unfair, however, since there have actually been quite a few people involved in seeing these results and reaching these conclusions. This seems like a good place to thank these people.

At the risk of being vague, it suffices to say that my attention has been divided and detracted by various diversions, academic and otherwise. I owe a huge debt of gratitude, not to mention a large number of cappucinos, to Professor Martin Vetterli for helping me put and keep a focus on my work. His guidance, regardless of geography, has been invaluable, and collaborating with him has been both a privilege and a joy.

I would also like to thank Professor Edward Lee for his unflinching support of my research. He opened the door to this world of possibilities and gave me the freedom to explore it at will. In addition, his standards of quality and clarity have been nothing short of inspirational.

Furthermore, I would like to thank the people who introduced me to the field of computer music, namely David Wessel, Adrian Freed, and Mark Goldstein; along the same lines, I am grateful to Xavier Rodet and Julius Smith for their assistance in formalizing and resolving a number of open issues. I would also like to thank Mark Dolson, Gianpaolo Evangelista, Jean Laroche, Michael Lee, Scott Levine, Brian Link, Dana Massey, Alan Peevers, and Tom Quatieri for their interest in my work and for various engaging conversations along the way. In addition, I would like to express my appreciation to Gary Elko for setting this wheel in motion in the first place, and to Heather Brown (*née*), Ruth Gjerde, Mary Stewart, and Christopher Hylands for helping to keep the wheel moving.

I would like to generally thank both the Ptolemy group and the wavelet group for their camaraderie; specifically, I would like to thank Vivek Goyal for his impeccability, Paolo Prandoni for his optimally tuned cynicism, and Jeff Carruthers, Grace Chang, and Matt Podolsky for their various roles in all this. Finally, it would be a drastic omission if this account did not pay proper homage to the people that have helped me shuffle through the everyday stuff; thanks especially to Joe, Luisa, Paul, Brian, Colin, Dave, and, last but most of all, my family.

At the risk of being melodramatic, I am inclined to mention that at this late hour I am reminded of a Kipling quote that my parents paraphrased a while ago, something like: *If you can give sixty seconds worth for every minute, yours is the earth...* Frankly, I don't know if this is true since some of my minutes haven't been worth much at all. On the other hand, though, many of my days have had thirty-some hours, and maybe in the end it all averaged out fairly well. Anyway, thanks again to everyone who either had a hand in the convergence of this work or who helped me keep everything else from diverging too severely.

# Chapter 1

# Signal Models and Analysis-Synthesis

The term *signal modeling* refers to the task of describing a signal with respect to an underlying structure – a model of the signal's fundamental behavior. *Analysis* is the process of fitting such a model to a particular signal, and *synthesis* is the process by which a signal is reconstructed using the model and the analysis data. This chapter discusses the basic theory and applications of signal models, especially those in which a signal is represented as a weighted sum of simple components; such models are the focus of this thesis. For the most part, the models to be considered are tailored for application to audio signals; in anticipation of this, examples related to audio are employed throughout the introduction to shed light on general modeling issues.

## 1.1 Analysis-Synthesis Systems

Signal modeling methods can be interpreted in the conceptual framework of analysis-synthesis. A general analysis-synthesis system for signal modeling is shown in Figure 1.1. The analysis block derives data pertaining to the signal model; this data is used by the synthesis block to construct a signal estimate. When the estimate is not perfect, the difference between the original $x[n]$ and the reconstruction $\hat{x}[n]$ is nonzero; this difference signal $r[n] = x[n] - \hat{x}[n]$ is termed the *residual.* The analysis-synthesis framework for signal modeling is developed further in the following sections.

### 1.1.1 Signal Representations

A wide variety of models can be cast into the analysis-synthesis framework of Figure 1.1. Two specific cases that illustrate relevant issues will be considered here: filter banks and physical models.

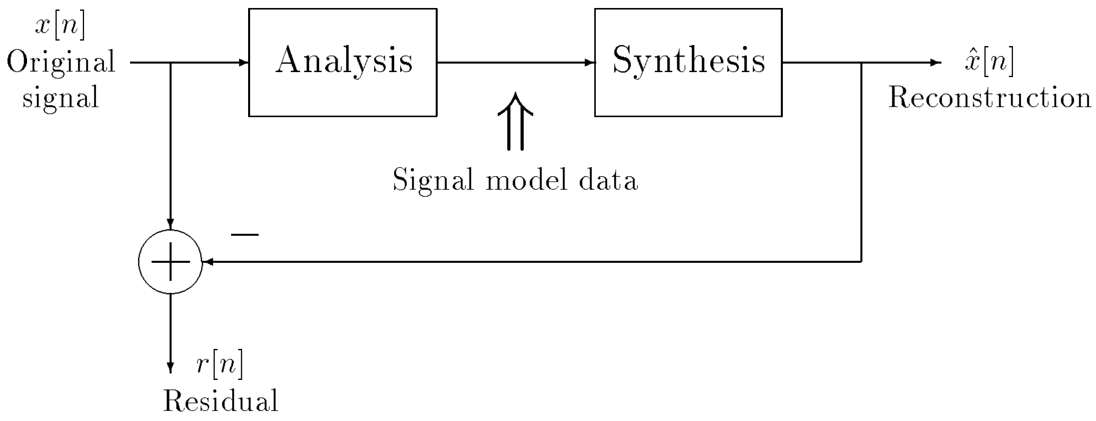

FIGURE 1.1: An analysis-synthesis framework for signal modeling. The analysis block derives the model data for the signal $x[n]$; the synthesis block constructs a signal estimate $\hat{x}[n]$ based on the analysis data. If the reconstruction is not perfect, there is a nonzero residual $r[n]$.

### Filter banks

A common approach to signal modeling involves using analysis and synthesis blocks consisting of filter banks. In such methods, the signal model consists of the subband signals derived by the analysis bank plus a description of the synthesis filter bank; reconstruction is carried out by applying the subband signals to the synthesis filters and accumulating their respective outputs. This filter bank scenario has been extensively considered in the literature. A few examples of filter bank techniques are short-time Fourier transforms [1], discrete wavelet transforms [2], discrete cosine transforms [3], lapped orthogonal transforms [4], and perceptual coding schemes wherein the filter bank is designed to mimic or exploit the properties of the human auditory or visual systems [5, 6, 7, 8, 9, 10, 11]. Such filter-based techniques have been widely applied in audio and image coding [7, 8, 12, 13, 14, 15, 16, 17, 18, 19], and a wide variety of designs and structures for analysis-synthesis filter banks have been proposed [2, 20].

### Physical models

A significantly different situation arises in the case of *physical modeling* of musical instruments [21, 22], which is a generalization of the source-filter approaches that are commonly used in speech processing applications [23, 6, 24, 25, 26, 27]. In source-filter approaches, the analysis consists of deriving a filter and choosing an appropriate source such that when the filter is driven by the source, the output is a reasonable estimate of the original signal; in some speech coding algorithms, the source mimics a glottal excitation while the filter models the shape of the vocal tract, meaning that the source-filter structure is designed to mirror the actual underlying physical system from which the speech signal originated. In physical modeling, this idea is extended to the case of arbitrary instruments, where both linear and nonlinear processing is essential to model the physical system [21]. Here, the purpose of the analysis is to derive a general physical description of the instrument in question. That physical description, which constitutes the signal model data in this case, is used to construct a synthesis system that mimics the instrument's behavior. In a guitar model, for instance, the model parameters derived by the analysis include the length, tension, and various wave propagation characteristics of the strings, the acoustic resonances of the guitar body, and the transfer properties of the string–body coupling. These physical parameters can be used to build a system that, when driven by a modeled excitation such as a string pluck, synthesizes a realistic guitar sound [21, 28, 29, 30].

### Mathematical and physical models

In either of the above cases, the signal model and the analysis-synthesis process are inherently connected: in the filter bank case, the signal is modeled as an aggregation of subbands; in a physical model, the signal is interpreted as the output of a complex physical system. While these representations are significantly different, they share a common conceptual framework in that the synthesis is driven by data from the analysis, and in that both the analysis and synthesis are carried out in accordance with an underlying signal model.

In the literature, physical models and signal models are typically differentiated. The foundation for this distinction is that physical models are concerned with the systems that are responsible for generating the signal in question, whereas signal models, in the strictest sense, are purely concerned with a mathematical approximation of the signal irrespective of its source – the signal is not estimated via an approximation of the generating physical system. As suggested in the previous sections, this differentiation, however, is somewhat immaterial; both approaches provide a representation of a signal in terms of a model and corresponding data. Certainly, physical models rely on mathematical analysis; furthermore, mathematical models are frequently based on physical considerations. While the models examined in this thesis are categorically mathematical, in each case the representation is supported by underlying physical principles, *e.g.* pitch periodicity.

### Additive models

The general topic of this thesis is mathematical signal modeling; as stated above, the models are improved by physical insights. The designation of a model as *mathematical* is rather general, though. More specifically, the focus of this thesis is additive signal models of the form

$$x[n] \;=\; \sum_{i=1}^{I} \alpha_i g_i[n], \tag{1.1}$$

wherein a signal is represented as a weighted sum of basic components; such models are referred to as *decompositions* or *expansions*. Of particular interest in these types of models is the capability of *successive refinement*. As will be seen, modeling algorithms can be designed such that the signal approximation is successively improved as the number of elements in the decomposition is increased; the improvement is measured using a metric such as mean-squared error. This notion suggests another similarity between mathematical and physical models; in either case, the signal estimate is improved by making the model more complex – either by using a more complicated physical model or by using more terms in the expansion. In this light, the advantage of additive models is that the model enhancement is carried out by relatively simple mathematics rather than complicated physical analyses as in the physical modeling case.

Signal models of the form given in Equation (1.1) are traditionally grouped into two categories: *parametric* and *nonparametric*. The fundamental distinction is that in nonparametric methods, the components $g_i[n]$ are a fixed function set, such as a basis; standard transform coders, for instance, belong to this class. In parametric methods, on the other hand, the components are derived using parameters extracted from the signal. These issues will be discussed further throughout this thesis; for instance, it will be shown in Chapter 6 that the inherent signal-adaptivity of parametric models can be achieved in models that are nonparametric according to this definition. In other words, for some types of models the distinction is basically moot.

General additive models have been under consideration in the field of computer music since its inception [31, 32, 33, 34, 35]. The basic idea of such *additive synthesis* is that a complex sound can be constructed by accumulating a large number of simple sounds. This notion is essential to the task of modeling musical signals; it is discussed further in the section on *granular synthesis* (Section 1.5.4) and is an underlying theme of this thesis.

### 1.1.2 Perfect and Near-Perfect Reconstruction

Filter banks satisfying *perfect reconstruction* constraints have received considerable attention in the literature [2, 20]. The term "perfect reconstruction" was coined to describe analysis-synthesis filter banks where the reconstruction is an exact duplicate of the original, with the possible exception of a time delay and a scale factor:

$$\hat{x}[n] \;=\; Ax[n-\delta]. \tag{1.2}$$

This notion, however, is by no means limited to the case of filter bank models; any model that meets the above requirement can be classified as a perfect reconstruction approach.

Throughout, $A = 1$ and $\delta = 0$ will often be assumed without loss of generality.

In perfect reconstruction systems, provided that the gain and delay are compensated for, the residual signal indicated in Figure 1.1 is uniformly zero. In practice, however, perfect reconstruction is not generally achievable; in the filter bank case, for instance, subband quantization effects and channel noise interfere with the reconstruction process. Given these inherent difficulties with implementing perfect reconstruction systems, the design of near-perfect reconstruction systems has been considered for filter bank models as well as more general cases. In these approaches, the models are designed such that the reconstruction error has particular properties; for instance, filter banks for audio coding are typically formulated with the intent of using auditory masking principles to render the reconstruction error imperceptible [7, 9, 6, 10, 11].

As stated, signal models typically cannot achieve perfect reconstruction. This is particularly true in cases where the representation contains less data than the original signal, *i.e.* in cases where *compression* is achieved. Beyond those cases, some models, regardless of compression considerations, simply do not account for perfect reconstruction. In audiovisual applications, these situations can be viewed in light of a looser near-perfect reconstruction criterion, that of *perceptual losslessness* or *transparency*, which is achieved in an analysis-synthesis system if the reconstructed signal is perceptually equivalent to the original. Note that a perceptually lossless system typically invokes psychophysical phenomena such as masking to effect data reduction or compression; its signal representation may be more efficient than that of a perfect reconstruction system.

The notion of perceptual losslessness can be readily interpreted in terms of the analysis-synthesis structure of Figure 1.1. For one, a perfect reconstruction system is clearly lossless in this sense. In near-perfect models, however, to achieve perceptual losslessness it is necessary that either the analysis-synthesis residual contain only components that would be perceptually insignificant in the synthesis, or that the residual be modeled separately and reinjected into the reconstruction. The latter case is most general.

As will be demonstrated in Chapter 2, the residual characteristically contains signal features that are not well-represented by the signal model, or in other words, components that the analysis is not designed to identify and that the synthesis is not capable of constructing. If these components are important (perceptually or otherwise) it is necessary to introduce a distinct model for the residual that can represent such features appropriately. Such signal-plus-residual models have been applied to many signal processing problems; this is considered further in Chapter 4.

The signal models discussed in this thesis are generally near-perfect reconstruction approaches tailored for audio applications. For the sake of compression or data reduction, perceptually unimportant information is removed from the representation. Thus, it is necessary to incorporate notions of perceptual relevance in the models. For music, it is well-known that high-quality synthesis requires accurate reproduction of note onsets or *attacks* [7, 36]. This so-called *attack problem* will be addressed in each signal model; it provides a foundation for assessing the suitability of a model for musical signals. For approximate models of audio signals, the distortion of attacks, often described using the term *pre-echo*, leads to a visual cue for evaluating the models; comparative plots of original and reconstructed attacks are a reliable indicator of the relative auditory percepts.

Issues similar to the attack problem commonly arise in signal processing applications. In many analysis-synthesis scenarios, it is important to accurately model specific signal features; other features are relatively unimportant and need not be accurately represented. In other words, the reconstruction error measure depends on the very nature of the signal and the applications of the representation. One example of this is compression of ambulatory electrocardiogram (ECG) signals for future off-line analysis; for this purpose it is only important to preserve a few key features of the heartbeat signal, and thus high compression rates can be achieved [37].

## 1.2 Compact Representations

Two very different models were discussed in Section 1.1.1, namely filter bank and physical models. These examples suggest the wide range of modeling techniques that exist; despite this variety, a few general observations can be made. Any given model is only useful inasmuch as it provides a signal description that is pertinent to the application at hand; in general, the usefulness of a model is difficult to assess without *a priori* knowledge of the signal. Given an accurate model, a reasonable metric for further evaluation is the compaction of the representation that the model provides. If a representation is both accurate and compact, *i.e.* is not data intensive, then it can be concluded that the representation captures the primary or meaningful signal behavior; a compact model in some sense extracts the coherent structure of a signal [38, 39]. This insight suggests that accurate compact representations are applicable to the tasks of compression, denoising, analysis, and signal modification; these are discussed in turn.

### 1.2.1 Compression

It is perhaps obvious that by definition a compact representation is useful for compression. In terms of the additive signal model of Equation (1.1), a compact representation is one in which only a few of the model components $\alpha_i g_i[n]$ are significant. With regards to accurate waveform reconstruction, such compaction is achieved when only a few coefficients $\alpha_i$ have significant values, provided of course that the functions $g_i[n]$ all have the same norm. Then, negligible components can be thresholded, *i.e.* set to zero, without substantially degrading the signal reconstruction. In scenarios where perceptual criteria are relevant in determining the quality of the reconstruction, principles such as auditory masking can be invoked to achieve compaction; in some cases, masking phenomena can be used to justify neglecting components with relatively large coefficients.

Various algorithms for computing signal expansions have focused on optimizing compaction metrics such as the entropy or $L_1$ norm of the coefficients or the rate-distortion performance of the representation; these approaches allow for an exploration of the tradeoff between the amount of data in the representation and its accuracy in modeling the signal [40, 41, 42, 43]. In expansions where the coefficients are all of similar value, thresholding is not useful and compaction cannot be readily achieved; this issue will come up again in Section 1.4 and Chapter 6. Note that for the remainder of this thesis the terms compression and compaction will for the most part be used interchangeably.

### 1.2.2 Denoising

It has been argued that compression and denoising are linked [44]. This argument is based on the observation that white noise is essentially incompressible; for instance, an orthogonal transform of white noise is again white, *i.e.* there is no compaction in the transform data and thus no compression is achievable. In cases where a coherent signal is degraded by additive white noise, the noise in the signal is not compressible. Then, a compressed representation does not capture the noise; it extracts the primary structure of the signal and a reconstruction based on such a compact model is in some sense a denoised or *enhanced* version of the original. In cases where the signal is well-modeled as a white noise process and the degradations are coherent, *e.g.* digital data with a sinusoidal jammer, this argument does not readily apply.

In addition to the filter-based considerations of [44], the connection between compression and denoising has been explored in the Fourier domain [45] and in the wavelet domain [46]. In these approaches, the statistical assumption is that small expansion coefficients correspond to noise instead of important signal features; as a result, thresholding the coefficients results in denoising. There are various results in the literature for thresholding wavelet-based representations [46]; such approaches have been applied with some success to denoising old sound recordings [47, 48]. Furthermore, motivated by the observation that quantization is similar to a thresholding operation, there have been recent considerations of quantization as a denoising approach [49].

It is interesting to note that denoising via thresholding has an early correspondence in time-domain speech processing for dereverberation and removing background noise [50, 51]. In that method, referred to as *center-clipping*, a signal is set to zero if it is below a threshold; if it is above the threshold, the threshold is subtracted. For a threshold $a$, the center-clipped signal is

$$\hat{x}[n] \;=\; \begin{cases} x[n]-a & x[n]>a \\ 0 & x[n]<a, \end{cases} \tag{1.3}$$

which corresponds to soft-thresholding the signal in the time domain rather than in a transform domain as in the methods discussed above.[1] This approach was considered effective for removing long-scale reverberation, *i.e.* echoes that linger after the signal is no longer present; such reverberation decreases the intelligibility of speech. Furthermore, center-clipping is useful as a front end for pitch detection of speech and audio signals [1, 53]. The recent work in transform-domain thresholding can be viewed as an extension of center-clipping to other representations.

### 1.2.3 Analysis, Detection, and Estimation

In an accurate compact representation, the primary structures of the signal are well-modeled. Given the representation, then, it is possible to determine the basic behavior of the signal. Certain patterns of behavior, if present in the signal, can be clearly identified in the representation, and specific parameters relating to that behavior can be extracted from the model. In this light, a compact representation enables signal analysis and characterization as well as the related tasks of detection, identification, and estimation.

### 1.2.4 Modification

In audio applications, it is often desirable to carry out modifications such as time-scaling, pitch-shifting, and cross-synthesis. *Time-scaling* refers to altering the duration of a sound without changing its pitch; *pitch-shifting*, inversely, refers to modifying the perceived pitch of a sound without changing its duration. Finally, *cross-synthesis* is the process by which two sounds are merged in a meaningful way; an example of this is applying a guitar string excitation to a vocal tract filter, resulting in a "talking" guitar [54]. These modifications cannot be carried out flexibly and effectively using commercially available systems such as samplers or frequency-modulation (FM) synthesizers [55]. For this reason, it is of interest to explore the possibility of carrying out modifications based on additive signal models.

A signal model is only useful with regard to musical modifications if it identifies musically relevant features of the signal such as pitch and harmonic structure; thus, a certain amount of analysis is a prerequisite to modification capabilities. Furthermore, data reduction is of significant interest for efficient implementations. Such compression can be achieved via the framework of perceptual losslessness; the signal model can be simplified by exploiting the principles of auditory perception and masking. This simplification, however, can only be carried out if the model components can individually be interpreted in terms

[1] Note that this kind of thresholding nonlinearity does not necessarily yield objectionable perceptual artifacts in speech signals; a similar nonlinearity has been successfully applied in the recent literature to improve the performance of stereo echo cancellation without degrading the speech quality [52].

of perceptually relevant parameters. If the components are perceptually motivated, their structure can be modified in perceptually predictable and meaningful ways. Thus, a compact transparent representation in some sense has inherent modification capabilities. Given this interrelation of data reduction, signal analysis, and perceptual considerations, it can be concluded from the preceding discussions that the modification capabilities of a representation hinge on its compactness.

## 1.3 Parametric Methods

As discussed in Section 1.1, signal models have been traditionally categorized as *parametric* or *nonparametric*. In nonparametric methods, the model is constructed using a rigid set of functions whereas in parametric methods the components are based on parameters derived by analyzing the signal. Examples of parametric methods include source-filter and physical models [27, 21], linear predictive and prototype waveform speech coding [23, 56], granular analysis-synthesis of music [33], and the sinusoidal model [57, 36]. The sinusoidal model is discussed at length in Chapter 2; granular synthesis is described in Section 1.5.4. The other models are discussed to varying extents throughout this text.

The distinction between parametric and nonparametric methods is admittedly vague. For instance, the indices of the expansion functions in a nonparametric approach can be thought of as parameters, so the terminology is clearly somewhat inappropriate. The issue at hand is clarified in the next section, in which various nonparametric methods are reviewed, as well as in Chapter 2 in the treatment of the phase vocoder, where a nonparametric method is revamped into a parametric method to enable signal modifications and reliable synthesis. The latter discussion indicates that the real issue is one of signal adaptivity rather than parametrization, *i.e.* a description of a signal is most useful if the associated parameters are signal-adaptive. It should be noted that traditional signal-adaptive parametric representations are not generally capable of perfect reconstruction; this notion is revisited in Chapter 6, which presents signal-adaptive parametric models that can achieve perfect reconstruction in some cases. As will be discussed, such methods illustrate that the distinction between parametric and nonparametric is basically insubstantial.

## 1.4 Nonparametric Methods

In contrast to parametric methods, *nonparametric methods* for signal expansion involve expansion functions that are in some sense rigid; they cannot necessarily be represented by physically meaningful parameters. Arbitrary *basis expansions* and *overcomplete expansions* belong to the class of nonparametric methods. The expansion functions in these cases are simply sets of vectors that span the signal space; they do not necessarily have an underlying structure. Note that these nonparametric expansions are tightly linked to the methods of linear algebra; the following discussion thus relies on matrix formulations.

### 1.4.1 Basis Expansions

For a vector space $V$ of dimension $N$, a *basis* is a set of $N$ linearly independent vectors $\{b_1, b_2, \ldots, b_N\}$. Linear independence implies that there is no nonzero solution $\{\gamma_n\}$ to the equation

$$\sum_{n=1}^{N} \gamma_n b_n \;=\; 0. \tag{1.4}$$

Then, the matrix

$$B \;=\; [b_1\; b_2\; \cdots\; b_N]\,, \tag{1.5}$$

whose columns are the basis vectors $\{b_n\}$, is invertible. Given the linear independence property, it follows that any vector $x \in V$ can be expressed as a unique linear combination of the form

$$x \;=\; \sum_{n=1}^{N} \alpha_n b_n. \tag{1.6}$$

In matrix notation, this can be written as

$$x \;=\; B\alpha, \tag{1.7}$$

where $\alpha = [\alpha_1\; \alpha_2\; \alpha_3\; \ldots\; \alpha_N]^T$. The coefficients of the expansion are given by

$$\alpha \;=\; B^{-1}x. \tag{1.8}$$

Computation of a basis expansion can also be phrased without reference to the matrix inverse $B^{-1}$; this approach is provided by the framework of biorthogonal bases, in which the expansion coefficients are evaluated by inner products with a second basis. After that discussion, the specific case of orthogonal bases is examined and some familiar examples from signal processing are considered.

It should be noted that the discussion of basis expansions in this section does not rely on the norms of the basis vectors, but that no generality would be lost by restricting the basis vectors to having unit norm. In later considerations, it will indeed prove important that all the expansion functions have unit norm.

#### Biorthogonal bases

Two bases $\{a_1, a_2, \ldots, a_N\}$ and $\{b_1, b_2, \ldots, b_N\}$ are said to be a pair of *biorthogonal* bases if

$$A^H B \;=\; I, \tag{1.9}$$

where $H$ denotes the conjugate transpose, $I$ is the $N \times N$ identity matrix and the matrices $A$ and $B$ are given by

$$A = [a_1 \ a_2 \ \cdots \ a_N] \quad \text{and} \quad B = [b_1 \ b_2 \ \cdots \ b_N]. \tag{1.10}$$

Equation (1.9) can be equivalently expressed in terms of the basis vectors as the requirement that

$$\langle a_i, b_j \rangle \ = \ a_i^H b_j \ = \ \delta[i-j]. \tag{1.11}$$

Such biorthogonal bases are also referred to as *dual* bases.

Given the relationship in Equation (1.9), it is clear that

$$A^H \ = \ B^{-1}. \tag{1.12}$$

Then, because the left inverse and right inverse of an invertible square matrix are the same [58], the biorthogonality constraint corresponds to

$$AB^H \ = \ I \quad \text{and} \quad BA^H \ = \ I. \tag{1.13}$$

This yields a pair of simple expressions for expanding a signal $x$ with respect to the biorthogonal bases:

$$\begin{aligned} x \ &= \ AB^H x \qquad &&= \ BA^H x \\ &= \ \sum_{n=1}^{N} \langle b_n, x \rangle a_n \ &&= \ \sum_{n=1}^{N} \langle a_n, x \rangle b_n. \end{aligned} \tag{1.14}$$

This framework of biorthogonality leads to flexibility in the design of wavelet filter banks [2]. Furthermore, biorthogonality allows for independent evaluation of the expansion coefficients, which leads to fast algorithms for computing signal expansions.

### Orthogonal bases

An *orthogonal* basis is a special case of a biorthogonal basis in which the two biorthogonal or dual bases are identical; here, the orthogonality constraint is

$$\langle b_i, b_j \rangle \ = \ \delta[i-j], \tag{1.15}$$

which can be expressed in matrix form as

$$B^H B \ = \ I \ \Longrightarrow \ B^H \ = \ B^{-1}. \tag{1.16}$$

Strictly speaking, such bases are referred to as *orthonormal* bases [58]; however, since most applications involve unit-norm basis functions, there has been a growing tendency in the literature to use the terms orthogonal and orthonormal interchangeably [2].

For an expansion in an orthogonal basis, the coefficients for a signal $x$ are given by

$$\alpha \;=\; B^H x \quad\Longrightarrow\quad \alpha_n \;=\; \langle b_n, x\rangle, \tag{1.17}$$

so the expansion can be written as

$$x \;=\; \sum_{n=1}^{N} \langle b_n, x\rangle b_n. \tag{1.18}$$

As in the general biorthogonal case, the expansion coefficients can be independently evaluated.

### Examples of basis expansions

The following list summarily describes the wide variety of basis expansions that have been considered in the signal processing literature; supplementary details are supplied throughout the course of this thesis when needed:

- The *discrete Fourier transform* (DFT) involves representing a signal in terms of sinusoids. For a discrete-time signal of length $N$, the expansion functions are sinusoids of length $N$. Since the expansion functions do not have compact *time support*, *i.e.* none of the basis functions are time-localized, this representation is ineffective for modeling events with short duration. Localization can in some sense be achieved for the case of a purely periodic signal whose length is an integral multiple of the period $M$, for which a DFT of size $M$ provides an exact representation.

- The *short-time Fourier transform (STFT)* is a modification of the DFT that has improved time resolution; it allows for time-localized representation of transient events and similarly enables DFT-based modeling of signals that are not periodic. The STFT is carried out by segmenting the signal into frames and carrying out a separate DFT for each short-duration frame. The expansion functions in this case are sinusoids that are time-limited to the signal frame, so the representation of dynamic signal behavior is more localized than in the general Fourier case. This is examined in greater detail in Chapter 2 in the treatment of the phase vocoder and the progression of ideas leading to the sinusoidal model.

- *Block transforms.* This is a general name for approaches in which a signal is segmented into blocks of length $N$ and each segment is then decomposed in an $N$-dimensional basis. To achieve compression, the decompositions are quantized and thresholded, which leads to discontinuities in the reconstruction, *e.g.* blockiness in images and frame-rate distortion artifacts in audio. This issue is somewhat resolved by *lapped orthogonal transforms*, in which the support of the basis functions extends beyond the block boundaries, which allows for a higher degree of smoothness in approximate reconstructions [4, 59].

- *Critically sampled perfect reconstruction filter banks* compute expansions of signals with respect to a biorthogonal basis related to the impulse responses of the analysis and synthesis filters [2]. This idea is fundamental to recent signal processing developments such as wavelets and wavelet packets.

- *Wavelet packets* correspond to arbitrary iterations of two-channel filter banks [2]; such *iterated filter banks* are motivated by the observation that a perfect reconstruction model can be applied to the subband signals in a critically sampled perfect reconstruction filter bank without marring the reconstruction. This leads to arbitrary perfect reconstruction *tree-structured filter banks* and multiresolution capabilities as will be discussed in Section 1.5.1. Such trees can be made adaptive so that the filter bank configuration changes in time to adapt to changes in the input signal [60]; in such cases, however, the resulting model is no longer simply a basis expansion. This is discussed further in Section 1.4.2, Chapter 3.

- *The discrete wavelet transform* is a special case of a wavelet packet where the two filters are generally highpass and lowpass and the iteration is carried out successively on the lowpass branch. This results in an octave-band filter bank in which the sampling rate of a subband is proportional to its bandwidth. The resulting signal model is the *wavelet decomposition*, which consists of octave-band signal details plus a lowpass signal estimate given by the lowpass filter of the final iterated filter bank. This model generally provides significant compaction for images but not as much for audio [18, 19, 14, 15, 61]. As will be seen in Chapter 5, in audio applications it is necessary to incorporate adaptivity in wavelet-based models to achieve transparent compaction [14].

#### Shortcomings of basis expansions

Basis expansions have a serious drawback in that a given basis is not well-suited for decomposing a wide variety of signals. For any particular basis, it is trivial to provide examples for which the signal expansion is not compact; the uniqueness property of basis representations implies that a signal with a noncompact expansion can be constructed by simply linearly combining the $N$ basis vectors with $N$ weights that are of comparable magnitude.

Consider the well-known cases depicted in Figure 1.2. For the frequency-localized signal of Figure 1.2(a), the Fourier expansion shown in Figure 1.2(c) is appropriately sparse and indicates the important signal features; in contrast, an octave-band wavelet decomposition (Figure 1.2(e)) provides a poor representation because it is fundamentally

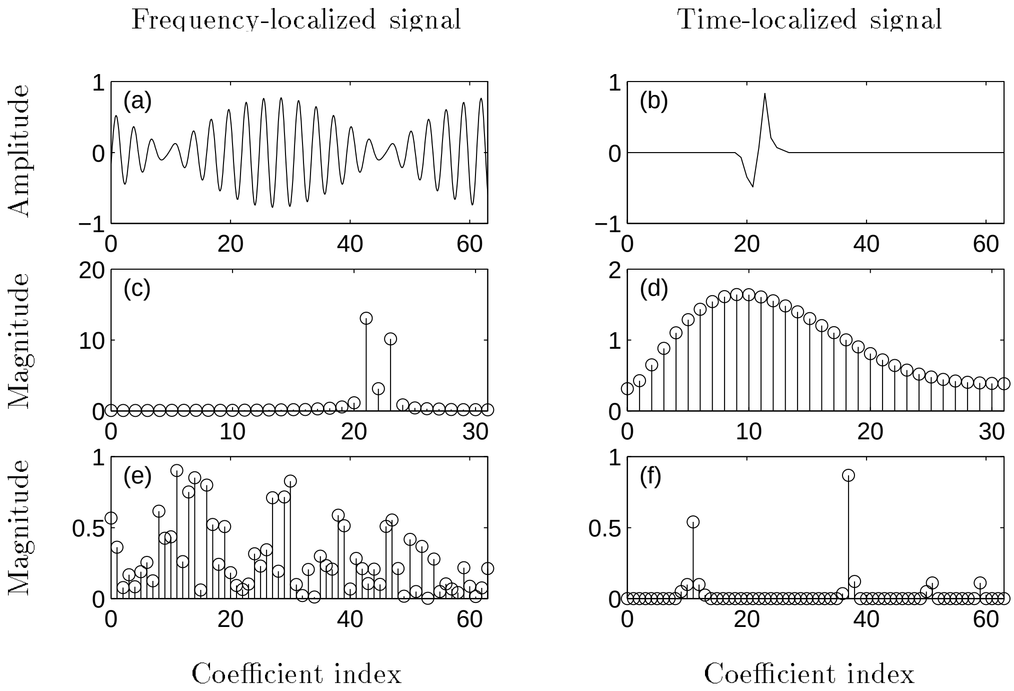

FIGURE 1.2: Shortcomings of basis expansions. The frequency-localized signal in (a) has a compact Fourier transform (c) and a noncompact wavelet decomposition (e); the time-localized signal in (b) has a noncompact Fourier expansion (d) and a compact wavelet representation (f).

unable to resolve multiple sinusoidal components in a single subband. For the time-localized signal of Figure 1.2(b), on the other hand, the Fourier representation of Figure 1.2(d) does not readily yield information about the basic signal structure; it cannot provide a compact model of a time-localized signal since none of the Fourier expansion functions are themselves time-localized. In this case, the wavelet transform (Figure 1.2(f)) yields a more effective signal model.

The shortcomings of basis expansions result from the attempt to represent arbitrary signals in terms of a very limited set of functions. Better representations can be derived by using expansion functions that are signal-adaptive; signal adaptivity can be achieved via parametric approaches such as the sinusoidal model [57, 36, 62], by using adaptive wavelet packets or best basis methods [40, 41, 60], or by choosing the expansion functions from an overcomplete set of time-frequency atoms [38]. These are fundamentally all examples of expansions based on an overcomplete set of vectors; this section focuses on the latter two, however, since these belong to the class of nonparametric methods. The term *overcomplete* means that the set or *dictionary* spans the signal space but includes more functions than is necessary to do so. Using a highly overcomplete dictionary of time-frequency atoms enables compact representation of a wide range of time-frequency behaviors; this depends however on choosing atoms from the dictionary that are appropriate for decomposing a given signal, *i.e.* the atoms are chosen in a signal-adaptive way. Basis expansions do not exhibit such signal adaptivity and as a result do not provide compact representations for arbitrary signals. According to the discussion in Section 1.2, this implies that basis expansions are not generally useful for signal analysis, compression, denoising, or modification. Such issues are revisited in Chapter 6; here, the shortcomings simply provide a motivation for considering overcomplete expansions.

### 1.4.2 Overcomplete Expansions

For a vector space $V$ of dimension $N$, a *complete* set is a set of $M$ vectors $\{d_1, d_2, \ldots, d_M\}$ that contains a basis ($M \geq N$). The set is furthermore referred to as *overcomplete* or *redundant* if in addition to a basis it also contains other distinct vectors ($M > N$). As will be seen, such redundancy leads to signal adaptivity and compact representations; algebraically, it implies that there are nonzero solutions $\{\gamma_m\}$ to the equation

$$\sum_{m=1}^{M} \gamma_m d_m \;=\; 0. \tag{1.19}$$

There are thus an infinite number of possible expansions of the form

$$x \;=\; \sum_{m=1}^{M} \alpha_m d_m. \tag{1.20}$$

Namely, if $\{\hat{\alpha}_m\}$ is a solution to the above equation and $\{\hat{\gamma}_m\}$ is a solution to Equation (1.19), then $\{\hat{\alpha}_m + \hat{\gamma}_m\}$ is also a solution:

$$x \;=\; \sum_{m=1}^{M} (\hat{\alpha}_m + \hat{\gamma}_m) d_m \;=\; \sum_{m=1}^{M} \hat{\alpha}_m d_m \;+\; \sum_{m=1}^{M} \hat{\gamma}_m d_m \;=\; \sum_{m=1}^{M} \hat{\alpha}_m d_m. \tag{1.21}$$

In matrix notation, with

$$D \;=\; [d_1 \; d_2 \; \cdots \; d_M], \tag{1.22}$$

Equation (1.20) can be written as

$$x \;=\; D\alpha, \tag{1.23}$$

where $\alpha = [\alpha_1 \; \alpha_2 \; \alpha_3 \; \ldots \; \alpha_M]^T$; the multiplicity of solutions can be interpreted in terms of the null space of $D$, which has nonzero dimension:

$$x \;=\; D(\hat{\alpha} + \hat{\gamma}) \;=\; D\hat{\alpha} + D\hat{\gamma} \;=\; D\hat{\alpha}. \tag{1.24}$$

Since there are many possible overcomplete expansions, there are likewise a variety of metrics and methods for computing the expansions. The overcomplete case thus lacks the structure of the basis case, where the coefficients of the expansion can be derived using an inverse matrix computation or, equivalently, correlations with a biorthogonal basis. As a result, the signal modeling advantages of overcomplete expansions come at the cost of additional computation.

### Derivation of overcomplete expansions

In the general basis case, the coefficients of the expansion are given by $\alpha = B^{-1}x$. For overcomplete expansions, one solution to Equation (1.20) can be found by using the singular value decomposition (SVD) of the dictionary matrix $D$ to derive its pseudo-inverse $D^{+}$. The coefficients $\alpha = D^{+}x$ provide a perfect model of the signal, but the model is however not compact; this is because the pseudo-inverse framework finds the solution $\alpha$ with minimum two-norm, which is a poor metric for compaction [58, 42].

Given this information about the SVD, not to mention the computational cost of the SVD itself, it is necessary to consider other solution methods if a compact representation is desired. There are two distinct approaches. The first class of methods involves structuring the dictionary so that it contains many bases; for a given signal, the *best basis* is chosen from the dictionary. The second class of methods are more general in that they apply to arbitrary dictionaries with no particular structure; here, the algorithms are especially designed to derive compact expansions. These are discussed briefly below, after an introduction to general overcomplete sets; all of these issues surrounding overcomplete expansions are discussed at length in Chapter 6.

### Frames

An overcomplete set of vectors $\{d_m\}$ is a *frame* if there exist two positive constants $E > 0$ and $F < \infty$, referred to as *frame bounds*, such that

$$E\|x\|^2 \;\leq\; \sum_m |\langle d_m, x\rangle|^2 \;\leq\; F\|x\|^2 \tag{1.25}$$

for any vector $x$. If $E = F$, the set is referred to as a *tight* frame and a signal can be expanded in a form reminiscent of the basis case:

$$x \;=\; \frac{1}{E}\sum_m \langle d_m, x\rangle d_m. \tag{1.26}$$

If the expansion vectors $d_m$ have unit norm, $E$ is a measure of the redundancy of the frame, namely $M/N$ for a frame consisting of $M$ vectors in an $N$-dimensional space.

The tight frame expansion in Equation (1.26) is equivalent to the expansion given by the SVD pseudo-inverse; it has the minimum two-norm of all possible expansions and thus does not achieve compaction. A similar expansion for frames that are not tight can be formulated in terms of a *dual frame*; it is also strongly connected to the SVD and does not lead to a sparse representation [2, 63].

More details on frames can be found in the literature [2, 63, 64]. It should simply be noted here that frames and oversampled filter banks are related in the same fashion as biorthogonal bases and critically sampled perfect reconstruction filter banks. Also, if a signal is to be reconstructed in a stable fashion from an expansion, meaning that bounded errors in the expansion coefficients lead to bounded errors in the reconstruction, it is necessary that the expansion set constitute a frame [2].

In the next two sections, two types of overcomplete expansions are considered. Fundamentally, these approaches are based on the theory of frames. Instead of using the terminology of frames, however, the discussions are phrased in terms of overcomplete dictionaries; it should be noted that these overcomplete dictionaries are indeed frames.

### Best basis methods

Best basis and adaptive wavelet packet methods, while not typically formalized in such a manner, can be interpreted as overcomplete expansions in which the dictionary contains a set of bases:

$$D = [B_1 \ B_2 \ B_3 \ \ldots \ ] . \tag{1.27}$$

For a given signal, the best basis from the dictionary is chosen for the expansion according to some metric such as the entropy of the coefficients [40], the mean-squared error of a thresholded expansion, a denoising measure [65, 66], or rate-distortion considerations [41, 60]. In each of the cited approaches, the bases in the dictionary correspond to tree-structured filter banks; there are thus mathematical relationships between the various bases and the expansions in those bases. In these cases, choosing the best basis (or wavelet packet) is equivalent to choosing the best filter bank structure, possibly time-varying, for a given signal. More general best basis approaches, where the various bases are not intrinsically related, have not been widely explored.

### Arbitrary dictionaries

As will be seen in the discussion of time-frequency resolution in Section 1.5.2, best basis methods involving tree-structured filter banks, *i.e.* adaptive wavelet packets, still have certain limitations for signal modeling because of the underlying structure of the sets of bases. While that structure does provide for efficient computation, in the task of signal modeling it becomes necessary to forego those computational advantages in order to provide for representation of arbitrary signal behavior. This suggestion leads to the more general approach of considering expansions in terms of arbitrary dictionaries and devising algorithms that find compact solutions. Such algorithms come in two forms: those that find exact solutions that maximize a compaction metric, either formally or heuristically [42, 67, 68], and those that find sparse approximate solutions that model the signal within some error tolerance [38, 39, 69]. These two paradigms have the same fundamental goal, namely compact modeling, but the frameworks are considerably different; in either case, however, the expansion functions are chosen in a signal-adaptive fashion and the algorithms for choosing the functions are decidedly nonlinear. This issue will be revisited in Chapter 6.

The various algorithms for deriving overcomplete expansions apply to arbitrary dictionaries. It is advantageous, however, if the dictionary elements can be parameterized in terms of relevant features such as time location, scale, and frequency modulation. Such parametric structure is useful for signal coding since the dictionaries and expansion functions can be represented with simple parameter sets, and for signal analysis in that the parameters provide an immediate indication of the signal behavior. Such notions appear throughout the entirety of this thesis. It is especially noteworthy at this point that using a parametric dictionary provides a connection between overcomplete expansions and parametric models; this connection will be discussed and exemplified in Chapter 6.

### 1.4.3 Example: Haar Functions

An illustrative comparison between basis expansions and overcomplete expansions is provided by a simple example involving *Haar functions*; these are the earliest and simplest examples of wavelet bases [2]. For discrete-time signals with eight time points, the matrix corresponding to a Haar wavelet basis with two scales is

$$B_{\text{Haar}} = \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix}^T, \tag{1.28}$$

where the basis consists of shifts by two and by four of the small scale and large scale Haar functions, respectively. The matrix is written in this transposed form to illustrate its relationship to the graphical description of the Haar basis given in Figure 1.3. An overcomplete Haar dictionary can be constructed by including all of the shifts by one of both small and large scales; the corresponding dictionary matrix is given in Figure 1.4.

Figure 1.5(a) shows the signal $x_1[n] = b_2$, the second column of the Haar basis matrix. Figure 1.5(b) shows a similar signal, $x_2[n] = x_1[n-1]$, a circular time-shift of $x_1[n]$. As shown in Figure 1.5(c), the decomposition of $x_1[n]$ in the Haar basis is compact – because $x_1[n]$ is actually in the basis; Figure 1.5(d), however, indicates that the Haar basis decomposition of $x_2[n]$ is not compact and is indeed a much less sparse model than the pure time-domain signal representation. Despite the strong relationship between the two signals, the transform representations are very different. The breakdown occurs in this particular example because the wavelet transform is not time-invariant; similar limitations apply to any basis expansion as discussed earlier. Expansions using

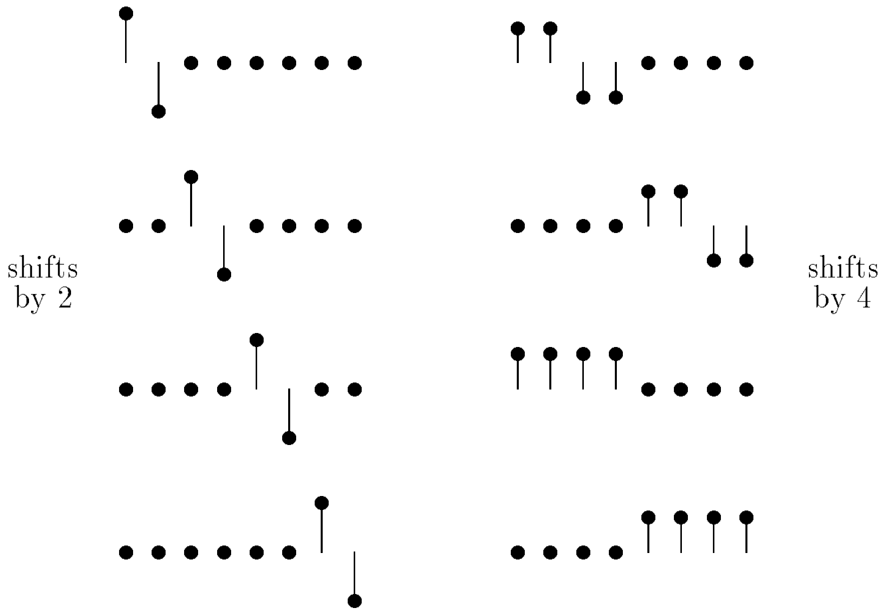

FIGURE 1.3: The Haar basis with two scales (for $C^8$).

the overcomplete Haar dictionary are shown in Figures 1.5(e) and 1.5(f). Both of these representations are compact. Noncompact overcomplete expansions derived using the SVD pseudo-inverse of $D_{\text{Haar}}$ are shown in Figures 1.5(g) and 1.5(h). Given the existence of the compact representations in Figures 1.5(e) and 1.5(f), the dispersion evident in the SVD signal models motivates the investigation of algorithms other than the SVD for deriving overcomplete expansions. Algorithms that derive compact expansions based on overcomplete dictionaries will be addressed in Chapter 6.

### 1.4.4 Geometric Interpretation of Signal Expansions

The linear algebra formulation developed above can be interpreted geometrically. Figure 1.6 shows a simple comparison of basis and overcomplete expansions in a two-dimensional vector space. The diagrams illustrate synthesis of the same signal using the vectors in an orthogonal basis, a biorthogonal basis, and an overcomplete dictionary, respectively; issues related to analysis-synthesis and modification are discussed below.

#### Analysis-synthesis

In each of the decompositions in Figure 1.6, the signal is reconstructed exactly as the sum of two expansion vectors. For the orthogonal basis, the expansion is unique and the expansion coefficients can be derived independently by simply projecting the signal onto

$$
D_{\text{Haar}} \quad = \quad \begin{bmatrix}
\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\
\frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & 0 & 0 & 0 & 0 \\
0 & \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & 0 & 0 & 0 \\
0 & 0 & \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & 0 & 0 \\
0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & 0 \\
0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} \\
\frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 \\
0 & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \\
0 & 0 & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 0 & 0 \\
0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 0 \\
0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2}
\end{bmatrix}^T
$$

FIGURE 1.4: The dictionary matrix for an overcomplete Haar set.

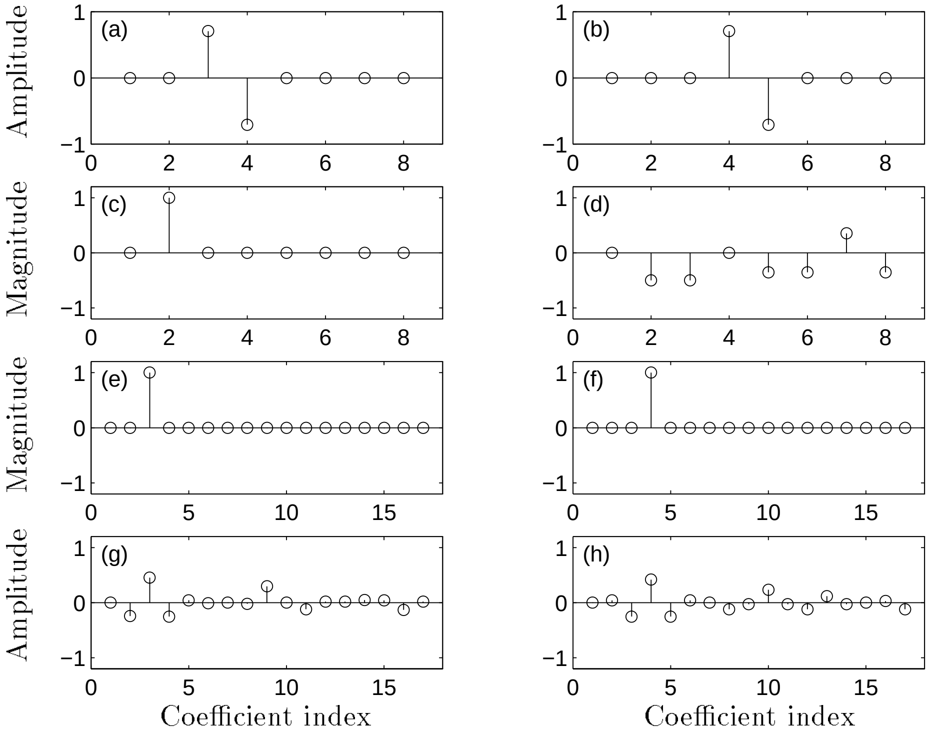

FIGURE 1.5: Comparison of decompositions in the Haar basis of Equation (1.28) and the Haar dictionary of Equation (1.29). Decompositions of signals (a) and (b) appear in the column beneath the respective signal. The basis expansion in (c) is compact, while that in (d) provides a poor model. The overcomplete expansions in (e) and (f) are compact, but these cannot generally be computed by linear methods such as the SVD, which for this case yields the noncompact expansions given in (g) and (h).

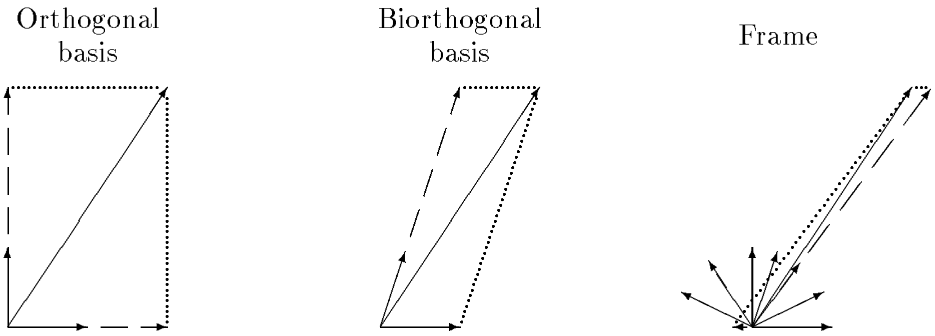

FIGURE 1.6: Geometric interpretation of signal expansions for orthogonal and biorthogonal bases and an overcomplete dictionary or frame.

the basis vectors. For the biorthogonal basis, the expansion vectors are not orthogonal; the expansion is still unique and the coefficients can still be independently evaluated, but the evaluation of the coefficients is done by projection onto a dual basis as described in Section 1.4.1. For the overcomplete frame, an infinite number of representations are possible since the vectors in the frame are linearly dependent. One way to compute such an overcomplete expansion is to project the signal onto a dual frame; such methods, however, are related to the SVD and do not yield compact models [70]. As discussed in Section 1.4.2, there are a variety of other methods for deriving overcomplete expansions. In this example, it is clear that a compact model can be achieved by using the frame vector that is most highly correlated with the signal since the projection of the signal onto this vector captures most of the signal energy. This greedy approach, known as *matching pursuit*, is explored further in Chapter 6 for higher-dimensional cases.

### Modification

Modifications based on signal models involve either adjusting the expansion coefficients, the expansion functions, or both. It is desirable in any of these cases that the outcome of the modification be predictable. In this section, the case of coefficient modification is discussed since the vector interpretation provided above lends immediate insight; modifying the coefficients simply amounts to adjusting the lengths of the component vectors in the synthesis. In the orthogonal case, the independence of the components leads to a certain robustness for modifications since each projection can be modified independently; if the orthogonal axes correspond to perceptual features to be adjusted, these features can be separately adjusted. In the biorthogonal case, to achieve the equivalent modification with respect to the orthogonal axes, the coupling between the projections must be taken into account. The most interesting caveat occurs in the frame case, however; because an overcomplete set is linearly dependent, some linear combinations of the frame vectors will add to zero. This means that some modifications of the expansion coefficients, namely those that correspond to adding vectors in the null space of the dictionary matrix $D$, will have no effect on the reconstruction. This may seem to be at odds with the previous assertion that compact models are useful for modification, but this is not necessarily the case. If fundamental signal structures are isolated as in compact models, the corresponding coefficients and functions can be modified jointly to avoid such difficulties. In Chapter 2, such issues arise in the context of establishing constraints on the synthesis components to avoid distortion in the reconstruction.

## 1.5 Time-Frequency Decompositions

The domains of *time* and *frequency* are fundamental to signal descriptions; relatively recently, *scale* has been considered as another appropriate domain for signal analysis [2]. These various arenas, in addition to being mathematically cohesive, are well-rooted in physical and perceptual foundations; generally speaking, the human perceptual experience can in some sense be well summarized in terms of when an event occurred (time), the duration of a given event (scale), and the rate of occurrence of events (frequency).

In this section, the notion of joint time-frequency representation of a signal is explored; the basic idea is that a model should indicate the local time and frequency behavior of a signal. Some extent of time localization is necessary for real-world processing of signals; it is impractical to model a signal defined over all time, so some time-localized or sequential approach to processing is needed. Time localization is also important for modeling transients in nonstationary signals; in arbitrary signals, various transients may have highly variable durations, so scale localization is also desirable in signal modeling. Finally, frequency localization is of interest because of the relationship of frequency to pitch in audio signals, and because of the importance of frequency in understanding the behavior of linear systems. Given these motivations, signal models of the form

$$x[n] \;\approx\; \sum_i \alpha_i g_i[n] \tag{1.29}$$

are of special interest when the expansion functions $g_i[n]$ are localized in time-frequency, since such expansions indicate the local time-frequency characteristics of a signal. Such cases, first elaborated by Gabor from both theoretical and psychoacoustic standpoints [71, 72], are referred to as time-frequency atomic decompositions; the localized functions $g_i[n]$ are time-frequency *atoms*, fundamental particles which comprise natural signals.

Atomic decompositions lead naturally to graphical time-frequency representations that are useful for signal analysis. Unfortunately, the resolution of any such analysis is fundamentally limited by physical principles [73, 74, 75]. This is the subject of Section 1.5.1, which discusses resolution tradeoffs between the various representation domains.

With these tradeoffs in mind, various methods for visualizing time-frequency models are discussed in Sections 1.5.2 and 1.5.3. Finally, time-frequency atomic decompositions have been of interest in the field of computer music for some time [33, 76, 77, 78]; this is discussed in Section 1.5.4.

### 1.5.1 Time-Frequency Atoms, Localization, and Multiresolution

The time-frequency localization of any given atom is constrained by a resolution limitation equivalent to the Heisenberg uncertainty principle of quantum physics [71, 74]. In short, good frequency localization can only be achieved by analyzing over a long period of time, so it comes at the expense of poor time resolution; similarly, fine time resolution does not allow for accurate frequency resolution. Note that analysis over a long period of time involves considering large scale signal behavior, and that analysis over short periods of time involves examining small scale signal behavior; furthermore, it is sensible to analyze for low frequency components over large scales since such components by definition do not change rapidly in time, and likewise high frequency components should be analyzed over short scales. The point here is simply that scale is necessarily intertwined in any notion of time-frequency localization. These tradeoffs between localization in time, frequency, and scale are the motivation of the wavelet transform and multiresolution signal decompositions [79, 2].

The localization of an atom can be depicted by a *tile* on the time-frequency plane, which is simply a rectangular section centered at some $(t_0, \omega_0)$ and having some widths $\Delta_t$ and $\Delta_\omega$ that describe where most of the energy of the signal lies [2]:

$$\Delta_t^2 = \int_{-\infty}^{\infty} (t - t_0)^2 \left| x(t - t_0) \right|^2 dt \tag{1.30}$$

$$\Delta_\omega^2 = \int_{-\infty}^{\infty} (\omega - \omega_0)^2 \left| X(\omega - \omega_0) \right|^2 d\omega. \tag{1.31}$$

The uncertainty principle provides a lower bound on the product of these widths:

$$\Delta_t \Delta_\omega \geq \sqrt{\frac{\pi}{2}}. \tag{1.32}$$

This uncertainty bound implies that there is a lower bound on the area of a time-frequency tile. It should be noted that non-rectangular tiles can also be formulated [80, 81, 82].

Within the limit of the resolution bound, many tile shapes are possible. These correspond to atoms ranging from impulses, which are narrow in time and broad in frequency, to sinusoids, which are broad in time and narrow in frequency; intermediate tile shapes basically correspond to modulated windows, *i.e.* time-windowed sinusoids. Various tiles are depicted in Figure 1.7.

It should be noted that tiles with area close to the uncertainty bound are of primary interest; larger tiles do not provide the desired localized information about the

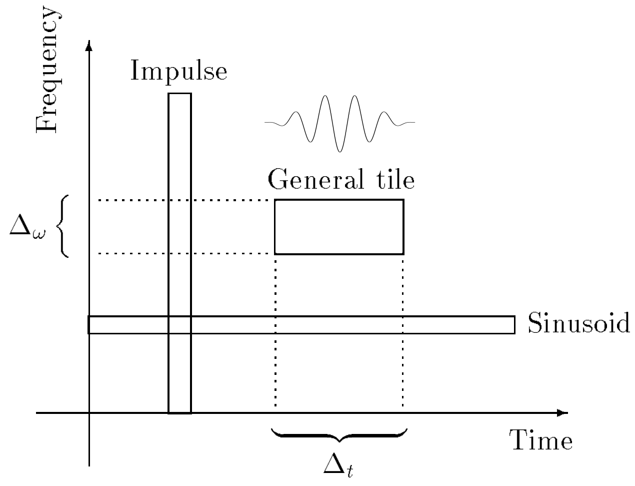

FIGURE 1.7: Tiles depicting the time-frequency localization of various expansion functions.

signal. With this in mind, one approach to generating a set of expansion functions for signal modeling is to start with a *mother* tile of small area and to derive a corresponding *family* of tiles, each having the same area, by scaling the time and frequency widths by inverse factors and allowing for shifts in time. Mathematically, this is given by

$$g_{a,b}(t) \;=\; \frac{1}{\sqrt{a}} g\left(\frac{t-b}{a}\right), \tag{1.33}$$

where $g(t)$ is the mother function. The continuous-time wavelet transform is based on families of this nature; restricting the scales and time shifts to powers of two results in the standard discrete-time wavelet transform. Expansion using such a set with functions with variable scale leads to a *multiresolution* signal model, which is physically sensible given the time-frequency tradeoffs discussed earlier.

Given a signal expansion in terms of a set of tiles, the signal can be readily modified by altering the underlying tiles. Time-shift, modulation, and scaling modifications of tiles are depicted in Figure 1.8. One caveat to note is that synthesis difficulties may arise if the tiles are modified in such a way that the synthesis algorithm is not capable of constructing the new tiles, *i.e.* if the new tiles are not in the signal model dictionary. This occurs in basis expansions; for instance, in the case of critically sampled filter banks, arbitrary modifications of the subband signals yield undesirable aliasing artifacts. The enhancement of modification capabilities is thus another motivation for using overcomplete expansions instead of basis expansions.

In this framework of tiles, the interpretation is that each expansion function in a decomposition analyzes the signal behavior in the time-frequency region indicated by its tile. Given that an arbitrary signal may have energy anywhere in the time-frequency plane, the objective of adaptive signal modeling is to decide where to place tiles to capture

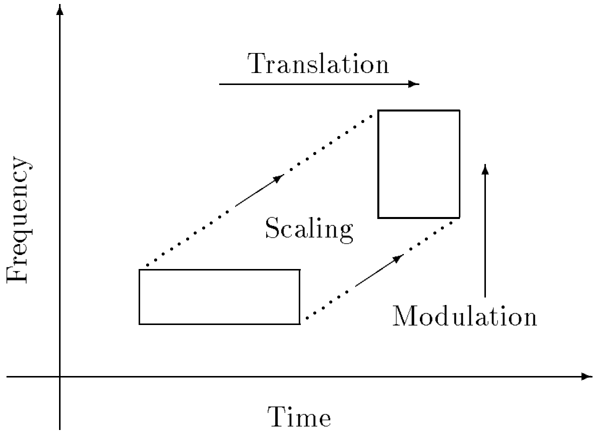

Figure 1.8: Modification of time-frequency tiles: translation, modulation, and scaling.

the signal energy. Tile-based interpretations of various time-frequency signal models are discussed in the next section.

### 1.5.2 Tilings of the Time-Frequency Plane

Signal expansions can be interpreted in terms of time-frequency tiles. For instance, a basis expansion for an $N$-dimensional signal can be visualized as a set of $N$ tiles that cover the time-frequency plane without any gaps or overlap. Examples of such time-frequency *tilings* are given in Figure 1.9; in visualizing an actual expansion, each tile is shaded to depict where the signal energy lies, *i.e.* to indicate the amplitude of the corresponding expansion function.

As indicated in Figure 1.9, the tilings for Fourier and wavelet transforms have regular structures; this equates to a certain simplicity in the computation of the corresponding expansion. As discussed in Section 1.4.1, however, these basis expansions have certain limitations for representing arbitrary signals. For that reason, it is of interest to consider tilings with more arbitrary structure. This is the idea in best basis and adaptive wavelet packet methods, where the best tiling for a particular signal is chosen; the best basis from a dictionary of bases is picked, according to some metric [40, 41, 60, 66, 65].

The time-varying tiling depicted in Figure 1.9 is intended as an example of an adaptive wavelet packet implemented with a signal-adaptive filter bank. This approach is suitable for a wide class of signals and allows for efficient computation, but the tiling is still restricted by the dyadic relationships between the scales, modulations, and time-shifts. The lack of complete generality arises because the tile sets under consideration cover the plane *exactly*; this captures all of the signal energy, but not necessarily in a

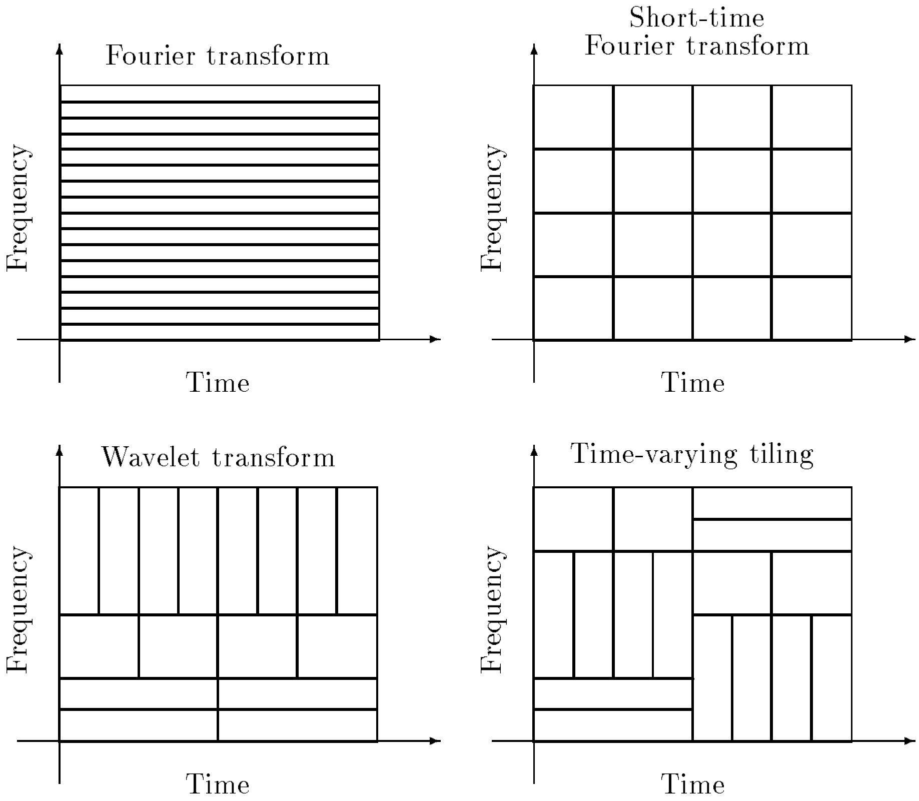

FIGURE 1.9: Tilings of the time-frequency plane for a Fourier transform, short-time Fourier transform, wavelet transform, and wavelet packet.

compact way. In the overcomplete case, overlapping tiles are admitted into the signal decomposition; compact models can then be achieved by choosing a few such general tiles that cover the regions in the time-frequency plane where the signal has significant energy.

### 1.5.3 Quadratic Time-Frequency Representations

Quadratic time-frequency representations or *bilinear expansions* have received considerable attention in the literature [83]. Fundamentally, such approaches are based on the Wigner-Ville distribution (WVD):

$$WVD\{x\}(\omega, \tau) \;=\; \int_{-\infty}^{\infty} x\left(\tau + \frac{t}{2}\right) x\left(\tau - \frac{t}{2}\right) e^{-j\omega t} dt. \tag{1.34}$$

Such representations provide improved resolution over linear expansions, but at the expense of the appearance of cross terms for signals with multiple components. For example, for a signal that consists of a single linear *chirp* (sinusoid with linearly increasing frequency), this behavior is clearly identifiable in the distribution; for a signal consisting of two crossing chirps, the product in the integral yields cross terms that degrade the *readability* of the time-frequency distribution [84, 85]. These cross-terms can be smoothed out in various ways, but always with the countereffect of decreasing the resolution of the signal representation [2, 86, 87].

Cross-terms detract from the usefulness of a quadratic time-frequency representation. In some sense, the cross-terms result in a noncompact model; they are extraneous elements in the representation that impede signal analysis. Even in cases where the cross-terms are smoothed out, the loss of resolution corresponds to a loss of compaction, so this problem with quadratic time-frequency representations is quite general. One approach is to improve the resolution of a smoothed representation by a nonlinear post-processing method referred to as *reallocation* or *reassignment*, in which the focus of the distribution is successively refined [85, 88]. Another approach is to derive an atomic decomposition of the signal, perhaps approximate, and then define a time-frequency representation (TFR) of the signal as a weighted sum of the time-frequency representations of the atoms [38]:

$$x[n] \;\approx\; \sum_i \alpha_i g_i[n] \tag{1.35}$$

$$TFR\{x\}(\omega, \tau) \;=\; \sum_i |\alpha_i|^2 \; WVD\{g_i\}(\omega, \tau). \tag{1.36}$$

There are no cross-terms in distributions derived in this manner [38, 89]; thus, another motivation for atomic time-frequency models is that they lead to clear visual descriptions of signal behavior. Of course, if the atomic decomposition is erroneous, the visual description will not be particularly useful.

### 1.5.4 Granular Synthesis

Granular synthesis is a technique in computer music which involves accumulating a large number of basic sonic components or *grains* to create a substantial acoustic event [33]. This approach is based on a theory of sound and perception that was first proposed by Gabor [72]; he suggested that any sound could be described using a quantum representation where each acoustic quantum or grain corresponds to a local time-frequency component of the sound. Such descriptions are psychoacoustically appropriate given the time-frequency resolution tradeoffs and limitations observed in the auditory system.

In early efforts in granular music synthesis, artificial sounds were composed by combining thousands of parameterized grains [33]. Individual grains were generated according to synthetic parameters describing both time-domain and frequency-domain characteristics, for example time location, duration, envelope shape, and modulation. This method was restricted to the synthesis of artificial sounds, however, because the representation paradigm did not have an accompanying analysis capable of deriving granular decompositions of existing natural sounds [78].

Simple analysis techniques for deriving grains from real sounds have been proposed in the literature [76, 77]. The objective of such *granulation* approaches is to derive a representation of natural sounds that enables modifications such as time-scaling or pitch-shifting prior to resynthesis. The basic idea in these methods is to extract grains by applying time-domain windows to the signal. Each windowed portion of the signal is treated as a grain, and parameterized by its window function and time location. These grains can be repositioned in time or resampled in various ways to achieve desirable signal modifications [76, 77]. Similar ideas have been explored in the speech processing community [56, 90].

Grains derived by the time-windowing process can be interpreted as signal-dependent expansion functions. If the grains are chosen judiciously, *e.g.* to correspond to pitch periods of a voiced sound, then the representation captures important signal structures and can as a result be useful for both coding and modification. Because of the complicated time structure of natural sounds, however, grains derived in this manner are generally difficult to represent efficiently and are thus not particularly applicable to signal coding. Nevertheless, this method is of interest because of its modification capabilities and its underlying signal adaptivity.

The time-windowed signal components derived by granulation are disparate from the fundamental acoustic quanta suggested by Gabor; time-windowing of the signal, while effective for modifications, is not an appropriate analysis for Gabor's time-frequency representation. With that as motivation, the three distinct signal models in this thesis are interpreted as granulation approaches: the sinusoidal model, pitch-synchronous expansions, and atomic models based on overcomplete time-frequency dictionaries can all be viewed in this light. These models provide time-frequency grains for additive synthetic reconstruction of natural signals, and these grains can generally be thought of as tiles on the time-frequency plane.

## 1.6 Overview

This thesis is concerned with signal models of the form given in Equation (1.1), namely additive expansions. The models in Chapters 2 through 5 can be classified as parametric approaches. On the other hand, Chapter 6 discusses a method that would be traditionally classified as nonparametric but which actually demonstrates that the distinction between the two types of models is artificial. A more detailed outline is given below, followed by a discussion of the themes of the thesis.

### 1.6.1 Outline

The contents of this thesis are as follows. First, Chapter 2 discusses the sinusoidal model, in which the expansion functions are time-evolving sinusoids. This approach is presented as an evolution of the nonparametric short-time Fourier transform into the phase vocoder and finally the fully parametric sinusoidal model; the chapter includes detailed treatments of the STFT, analysis for the sinusoidal model, and methods for sinusoidal synthesis. Chapter 3 provides an interpretation of the sinusoidal model in terms of time-frequency atoms, which motivates the consideration of multiresolution extensions of the model for accurately representing localized signal behavior. Chapter 4 discusses the sinusoidal analysis-synthesis residual and presents a perceptually motivated model for the residual signal. Chapter 5 examines pitch-synchronous frameworks for both sinusoidal models and wavelet transforms; the estimation of the pitch parameter is shown to provide a useful avenue for improving the signal representation in both cases. In Chapter 6, overcomplete expansions are revisited; signal modeling is interpreted as an inverse problem and connections between structured overcomplete expansions and parametric methods are considered. The chapter discusses the matching pursuit algorithm for computing overcomplete expansions, and considers overcomplete dictionaries based on damped sinusoids, for which expansions can be computed using simple recursive filter banks. Finally, Chapter 7 reviews the results of the thesis and presents various concluding remarks about adaptive signal models and related algorithms.

### 1.6.2 Themes

This thesis has a number of underlying and recurring themes. In a sense, this text is about the relationships between these themes. The basic conceptual framework of this thesis has been central to several preliminary presentations in the literature [91, 62], but in this document the various issues are explored in greater detail; furthermore, considerable attention is given to review of fundamental background material. The themes of this thesis are as follows.

### Filter banks and multiresolution

Filter bank theory and design appear in several places in this thesis. Primarily, the thesis deals with the interpretation of filter banks as analysis-synthesis structures for signal modeling. The connection between multirate filter banks and multiresolution signal modeling is explored.

### Signal-adaptive representations

Each of the signal models or representations discussed in this thesis exhibits signal adaptivity. In the sinusoidal and pitch-synchronous models, the decompositions are signal-adaptive in that the expansion functions are generated based on data extracted from the signal. In the overcomplete expansions, the models are adaptive in that the expansion functions for the signal decomposition are chosen from the dictionary in a signal-dependent fashion.

### Parametric models

The expansion functions in the sinusoidal and pitch-synchronous models are generated based on parameters derived by the signal analysis. Such parametric expansions, as discussed in Section 1.2, are useful for characterization, compression, and modification of signals. Overcomplete expansions can be similarly parametric in nature if the underlying dictionary has a meaningful parametric structure. In such cases, the traditional distinction between parametric and nonparametric methods evaporates, and the overcomplete expansion provides a highly useful signal model.

### Nonlinear analysis

In each model, the model estimation is inherently nonlinear. The sinusoidal and pitch-synchronous models rely on nonlinear parameter estimation and interpolation. The matching pursuit is inherently nonlinear in the way it selects the expansion functions from the overcomplete dictionary; it overcomes the inadequacies of linear methods such as the SVD while providing for successive refinement and compact sparse approximations. It has been argued that overcompleteness, when coupled with a nonlinear analysis, yields a signal-adaptive representation, so these notions are tightly coupled [92, 39].

### Atomic models

Finally, all of the models can be interpreted in terms of localized time-frequency atoms or grains. The notion of time-frequency decompositions has been discussed at length in several sections of this introduction, and will continue to play a major role throughout the remainder of this thesis.

# Chapter 2

# Sinusoidal Modeling

The sinusoidal model has been widely applied to speech coding and processing [57, 93, 94, 95, 96, 97, 98, 99] and audio analysis–modification–synthesis [36, 100, 101, 102, 103, 104, 105]. This chapter discusses the sinusoidal model, including analysis and synthesis techniques, reconstruction artifacts, and modification capabilities enabled by the parametric nature of the model. Time-domain and frequency-domain synthesis methods are examined. A thorough review of the short-time Fourier transform is included as an introduction to the discussion of the sinusoidal model.

## 2.1 The Sinusoidal Signal Model

A variety of sinusoidal modeling techniques have been explored in the literature [106, 96, 95, 98, 57, 36, 102, 101, 97]. These methods share fundamental common points, but also have substantial but sometimes subtle differences. For the sake of simplicity, this treatment adheres primarily to the approaches presented in the early literature on sinusoidal modeling [57, 36], and not on the many variations that have since been proposed [103, 97, 98]; comments on some other techniques such as [101, 107] are indeed included, but these inclusions are limited to techniques that are directly concerned with the modeling issues at hand. It should be noted that the issues to be discussed herein apply to sinusoidal modeling in general; their relevance is not limited by the adherence to the particular methods of [57, 36]. Also, note that the method of [102] is discussed at length in the section on frequency-domain synthesis, where various refinements are proposed.

### 2.1.1 The Sum-of-Partials Model

In sinusoidal modeling, a discrete-time signal $x[n]$ is modeled as a sum of evolving sinusoids called *partials*:

$$x[n] \approx \hat{x}[n] = \sum_{q=1}^{Q[n]} p_q[n] = \sum_{q=1}^{Q[n]} A_q[n] \cos \Theta_q[n], \tag{2.1}$$

where $Q[n]$ is the number of partials at time $n$. The $q$-th partial $p_q[n]$ has time-varying amplitude $A_q[n]$ and total phase $\Theta_q[n]$, which describes both its frequency evolution and phase offset. The additive components in the model are thus simply parameterized by amplitude and frequency functions or *tracks*. These tracks are assumed to vary on a time scale substantially longer than the sampling period, meaning that the parameter tracks can be reliably estimated at a subsampled rate. The assumption of slow variation leads to a compaction in the representation in the same way that downsampling of bandlimited signals leads to data reduction without loss of information.

The model in Equation (2.1) is reminiscent of the familiar Fourier series; the notion in Fourier series methods is that a periodic signal can be exactly represented by a sum of fixed harmonically related sinusoids. Purely periodic signals, however, are a mathematical abstract. Real-world oscillatory signals such as a musical note tend to be *pseudo-periodic*; they exhibit variations from period to period. The sinusoidal model is thus useful for modeling natural signals since it generalizes the Fourier series in the sense that the constituent sinusoids are allowed to evolve in time according to the signal behavior. Of course, the sinusoidal model is not limited to applications involving pseudo-periodic signals; models tailored specifically for pseudo-periodic signals will be discussed in Chapter 5.

Fundamentally, the sinusoidal model is useful because the parameters capture musically salient time-frequency characteristics such as spectral shape, harmonic structure, and loudness. Since it describes the primary musical information about the signal in a simple, compact form, the parameterization provides not only a reasonable coding representation but also a framework for carrying out desirable modifications such as pitch-shifting, time-scaling, and a wide variety of spectral transformations such as cross-synthesis [93, 94, 36, 102, 103, 108].

### 2.1.2 Deterministic-plus-Stochastic Decomposition

The approximation symbol in Equation (2.1) is included to imply that the sum-of-partials model does not provide an exact reconstruction of the signal. Since a sum of slowly-varying sinusoids is ineffective for modeling either impulsive events or highly uncorrelated noise, the sinusoidal model is not well-suited for representing broadband processes. As a result, the sinusoidal analysis-synthesis residual consists of such processes, which correspond to musically important signal features such as the colored breath noise in a flute sound or the impulsive mallet strikes of a marimba. Since these features are important for high-fidelity synthesis, an additional component is often included in the signal model to account for broadband processes:

$$x[n] \;=\; \hat{x}[n] \;+\; r[n] \;=\; d[n] \;+\; s[n]. \tag{2.2}$$

The resultant *deterministic-plus-stochastic* decomposition was introduced in [36, 100] and has been discussed in several later efforts [109, 110]. Using this terminology brings up salient issues about the theoretical distinction between deterministic and stochastic processes; to avoid such pitfalls, the following analogy is drawn: the deterministic part of the decomposition is likened to the sum-of-partials of Equation (2.1) and the stochastic part is similarly likened to the residual of the sinusoidal analysis-synthesis process, leading to a reconstruction-plus-residual decomposition. This method can then be considered in light of the conceptual framework of Chapter 1. The sinusoidal analysis-synthesis is described in Sections 2.3 to 2.6 from that, the characteristics of the residual are inferred, which leads to the residual modeling approach of Chapter 4.

## 2.2 The Phase Vocoder

Sinusoidal modeling can be viewed in a historical context as an evolution of short-time Fourier transform (STFT) and phase vocoder techniques. These methods and variations were developed and explored in a number of references [111, 112, 113, 114, 115, 116, 117, 118, 119]. In this treatment, the various ideas are presented in a progression which leads from the STFT to the phase vocoder; the shortcomings of these approaches serve to motivate the general sinusoidal model.

### 2.2.1 The Short-Time Fourier Transform

In this section, the STFT is defined and interpreted; it is shown that slightly revising the traditional definition leads to an alternative filter bank interpretation of the STFT that is appropriate for signal modeling. Perfect reconstruction constraints for such STFT filter banks are derived. In the literature, $z$-transform and matrix representations have been shown to be useful in analyzing the properties of such filter banks [2, 20, 120]. Here, for the sake of brevity, these methods are not explored; the STFT filter banks are treated using time-domain considerations.

**Definition of the short-time Fourier transform**

The short-time Fourier transform was described conceptually in Sections 1.4.1 and 1.5.1; basically, the goal of the STFT is to derive a time-localized representation of the frequency-domain behavior of a signal. The STFT is carried out by applying a sliding time window to the signal; this process isolates time-localized regions of the signal, which are each then analyzed using a discrete Fourier transform (DFT). Mathematically, this is given by

$$X[k,n] \;=\; \sum_{m=0}^{N-1} w[m]x[n+m]e^{-j\omega_k m}, \tag{2.3}$$

where the DFT is of size $K$, meaning that $\omega_k = 2\pi k/K$, and $w[m]$ is a time-domain window with zero value outside the interval $[0, N-1]$; windows with infinite time support have been discussed in the literature, but these will not be considered here [116]. In the early literature on time-frequency transforms, signal analysis-synthesis based on Gaussian windows was proposed by Gabor [71, 72]; given this historical foundation, the STFT is sometimes referred to as a Gabor transform [2].

The transform in Equation (2.3) can be expressed in a subsampled form which will be useful later:

$$X(k,i) \;=\; \sum_{m=0}^{N-1} w[m]x[m+iL]e^{-j\omega_k m}, \tag{2.4}$$

where $L$ is the analysis stride, the time distance between successive applications of the window to the data. The notation is as follows: brackets around the arguments are used to indicate a nonsubsampled STFT such as in $X[k,n]$, while parentheses are used to indicate subsampling as in $X(k,i)$, which is used in lieu of $X[k,iL]$ for the sake of neatness. Admittedly, the notation $X(k,i)$ is somewhat loose in that it does not incorporate the hop size, but to account for this difficulty the hop size of any subsampled STFTs under consideration will be indicated explicitly in the text. The subsampled form of the STFT is of interest since it allows for a reduction in the computational cost of the signal analysis and in the amount of data in the representation; it also affects the properties of the model and the reconstruction as will be demonstrated.

The definition of the STFT given in Equations (2.3) and (2.4) differ from that in traditional references on the STFT [111, 116, 115, 112], where the transform is expressed as

$$\tilde{X}[k,n] \;=\; \sum_{m=-\infty}^{\infty} \tilde{w}[n-m]x[m]e^{-j\omega_k m} \;=\; \sum_{m=n}^{n+N-1} \tilde{w}[n-m]x[m]e^{-j\omega_k m}, \tag{2.5}$$

or in subsampled form as

$$\tilde{X}(k,i) \;=\; \sum_{m=iL}^{iL+N-1} \tilde{w}[iL-m]x[m]e^{-j\omega_k m}, \tag{2.6}$$

where $\tilde{w}[m]$ is again a time-localized window. The range of $m$ in the sum, and hence the support of the window $\tilde{w}[n]$, is defined here in such a way that the transforms $X[k,n]$ and $\tilde{X}[k,n]$ refer to the same $N$-point segment of the signal and can thus be compared; it should be noted that in some treatments the STFT is expressed as in Equation (2.5) but without time-reversal of the window [20]. It will be shown that this reversal of the time index affects the interpretation of the transform as a filter bank; more importantly, however, the interpretation is affected by the time reference of the expansion functions. This latter issue is discussed below.

### The time reference of the STFT

In the formulation of the STFT in Equations (2.5) and (2.6), the expansion functions are sinusoids whose time reference is in some sense absolute; for different windowed signal segments, the expansion functions have the same time reference, $m = 0$, the time origin of the signal $x[m]$. On the other hand, in Equations (2.3) and (2.4) the time origin of the expansion functions is instead the starting point of the signal segment in question; the phases of the expansion coefficients for a segment refer to the time start of that particular segment. Note that the STFT can also be formulated such that the phase is referenced to the center of the time window, which is desirable in some cases [121]; this referencing is a straightforward extension that will play a role in sinusoidal modeling, but such phase-centering will not be used in the mathematical development of the STFT because of the slight complications it introduces.

The two formulations of the STFT have different interpretations with regards to signal modeling; this difference can be seen by relating the two STFT definitions [1, 112]:

$$\begin{aligned}
\tilde{X}[k,n] &= \sum_{m=n}^{n+N-1} \tilde{w}[n-m]x[m]e^{-j\omega_k m} \\
&= \sum_{m=0}^{N-1} \tilde{w}[-m]x[n+m]e^{-j\omega_k (m+n)} && \text{(change of index)} \\
&= e^{-j\omega_k n} \sum_{m=0}^{N-1} \tilde{w}[-m]x[n+m]e^{-j\omega_k m} \\
&= e^{-j\omega_k n} \sum_{m=0}^{N-1} w[m]x[n+m]e^{-j\omega_k m} && (\tilde{w}[m] = w[-m]) \\
\tilde{X}[k,n] &= e^{-j\omega_k n} X[k,n].
\end{aligned} \tag{2.7}$$

This formulation leads to two simple relationships:

$$X[k,n] = \tilde{X}[k,n]e^{j\omega_k n} \tag{2.8}$$

$$|X[k,n]| = |\tilde{X}[k,n]|. \tag{2.9}$$

The first expression affects the interpretation of the STFT as a filter bank; the time signal $X[k,n]$ is a modulated version of the baseband envelope signal $\tilde{X}[k,n]$, so the equivalent filter banks for the two cases will have different structures. The second expression plays a role in the interpretation of the STFT as a series of time-localized spectra; the short-time magnitude spectra are the same in either case. These relations have different consequences for sinusoidal modeling. First, magnitude considerations have no bearing because of the equivalence in Equation (2.9). On the other hand, because an estimate of the *local* phase of a partial is important for building a localized model of the original signal, Equation (2.8) indicates that $X[k,n]$ is a more useful representation for sinusoidal modeling than $\tilde{X}[k,n]$. This will become more apparent in Sections 2.3 and 2.4.

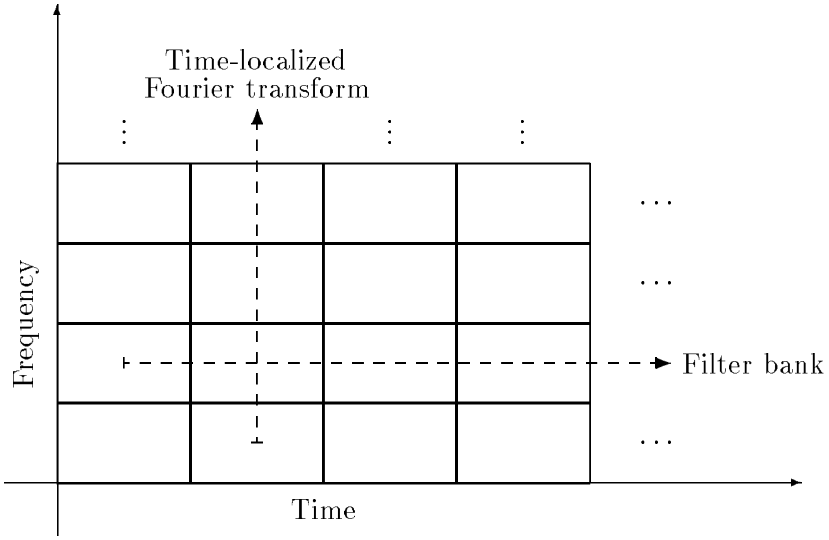

Figure 2.1: Interpretations of the short-time Fourier transform as a series of time-localized spectra (vertical) and as a bank of bandpass filters (horizontal).

### Interpretations of the STFT

In [111, 1] and other traditional treatments of the STFT, two interpretations are considered. First, the STFT can be viewed as a series of time-localized spectra; notationally, this corresponds to interpreting $X[k, n]$ as a function of frequency $k$ for a fixed $n$. Given that the derivation of a time-localized spectral representation was indeed the initial motivation of the STFT, the novelty lies in the second interpretation, where the STFT is viewed as a bank of bandpass filters. Here, $X[k, n]$ is thought of as a function of time $n$ for a fixed frequency $k$; it is simply the output of the $k$-th filter in the STFT filter bank. A depiction of these interpretations based on the time-frequency tiling of the STFT is given in Figure 2.1; indeed, the notion of a tiling unifies the two perspectives.

The two interpretations are discussed in the following sections; as will be seen, each interpretation provides a framework for signal reconstruction and each framework yields a perfect reconstruction constraint. In the traditional formulation of the STFT, the reconstruction constraints are different for the two interpretations, but can be related by a duality argument [111]. In the phase-localized formulation of Equations (2.3) and (2.4), the two frameworks immediately yield the same perfect reconstruction condition; this is not particularly surprising since the representation of the STFT as a time-frequency tiling suggests that a distinction between the two interpretations is indeed artificial. The mathematical details related to these issues are developed below; also, the differences in the signal models corresponding to the two STFT formulations are discussed.

### The STFT as a series of time-localized spectra

If the STFT is interpreted as a series of time-localized spectra, the accompanying reconstruction framework involves taking an inverse DFT (IDFT) of each local spectrum, and then connecting the resulting signal frames to synthesize the signal. If $K \geq N$, the IDFT simply returns the windowed signal segment:

$$\begin{aligned} \text{IDFT}\{X(k,i)\} &= w[m]x[m+iL] \quad \text{for} \quad 0 \leq m \leq N-1 \\ &= w[n-iL]x[n] \quad \text{for} \quad iL \leq n \leq iL+N-1, \end{aligned} \tag{2.10}$$

where the second step is carried out to simplify the upcoming formulation. Regarding the size of the DFT, when $K > N$ the DFT is *oversampled*, which results in a time-limited interpolation of the spectrum, which is analogous to the bandlimited interpolation that is characteristic of time-domain oversampling. The condition $K \geq N$ is imposed at this point to simplify the formulation; time-domain aliasing is introduced in the *undersampled* case $K < N$, meaning that the formulation must be revised to provide for time-domain aliasing cancellation [12]. The issue of time-domain aliasing cancellation is discussed in Section 2.2.2.

If the DFT is large enough that no aliasing occurs, reconstruction can be simply carried out by an *overlap-add* (OLA) process, possibly with a *synthesis* window, which will be denoted by $v[n]$ [116, 115, 112]:

$$\hat{x}[n] = \sum_i w[n-iL]v[n-iL]x[n]. \tag{2.11}$$

Perfect reconstruction is thus achieved if the windows $w[n]$ and $v[n]$ satisfy the constraint

$$\sum_i w[n-iL]v[n-iL] = 1 \tag{2.12}$$

or some other constant. This constraint is similar to but somewhat more general than the perfect reconstruction constraints given in [1, 111, 116, 115, 112]. Note that throughout this section the analysis and synthesis windows will both be assumed to be real-valued.

In cases where $v[n]$ is not explicitly specified, the synthesis window is equivalently a rectangular window covering the same time span as $w[n]$. For a rectangular synthesis window, the constraint in Equation (2.12) becomes

$$\sum_i w[n-iL] = 1. \tag{2.13}$$

The construction of windows with this property has been explored in the literature; a variety of *perfect reconstruction windows* have been proposed, for example rectangular and triangular windows and the Blackman-Harris family, which includes the familiar Hanning and Hamming windows [122, 123]. These are also referred to as windows with the *overlap-add property*, and will be denoted by $w_{\text{PR}}[n]$ in the following derivations. Note that *any* window function satisfies the condition in Equation (2.13) in the nonsubsampled case $L = 1$; note also that in the case $L = N$ the only window that has the overlap-add property is a rectangular window of length $N$. Functions that satisfy Equation (2.13) are also of interest for digital communication; the Nyquist criterion for avoiding intersymbol interference corresponds to a frequency-domain overlap-add property [124].

Windows that satisfy Equation (2.12) can be designed in a number of ways. The methods to be discussed rely on using familiar windows that satisfy (2.13) to jointly construct analysis and synthesis windows which satisfy (2.12); various analysis-synthesis window pairs designed in this way exhibit computational and modeling advantages [116, 115, 112, 102]. In one design approach, complementary powers of a perfect reconstruction window provide the analysis and synthesis windows:

$$\sum_i w_{\mathrm{PR}}[n-iL] \;=\; 1 \implies \sum_i \left(w_{\mathrm{PR}}[n-iL]\right)^c \left(w_{\mathrm{PR}}[n-iL]\right)^{1-c} \;=\; 1 \tag{2.14}$$

$$\implies \begin{cases} \text{Analysis window} & w[n] \;=\; \left(w_{\mathrm{PR}}[n]\right)^c \\ \text{Synthesis window} & v[n] \;=\; \left(w_{\mathrm{PR}}[n]\right)^{1-c}. \end{cases} \tag{2.15}$$

The case $c = \frac{1}{2}$, where the analysis and synthesis windows are equivalent, has been of some interest because of its symmetry. A second approach is as follows; given a perfect reconstruction window $w_{\mathrm{PR}}[n]$ and an arbitrary window $b[n]$ that is strictly nonzero over the time support of $w_{\mathrm{PR}}[n]$, the overlap-add property can be rephrased as follows:

$$\begin{aligned} \sum_i w_{\mathrm{PR}}[n-iL] \;=\; 1 \;\implies\; & \sum_i w_{\mathrm{PR}}[n-iL]\left(\frac{b[n-iL]}{b[n-iL]}\right) \;=\; 1 \\ \implies\; & \sum_i b[n-iL]\left(\frac{w_{\mathrm{PR}}[n-iL]}{b[n-iL]}\right) \;=\; 1 \end{aligned} \tag{2.16}$$

$$\implies \begin{cases} \text{Analysis window} & w[n] \;=\; b[n] \\ \text{Synthesis window} & v[n] \;=\; \dfrac{w_{\mathrm{PR}}[n]}{b[n]}. \end{cases} \tag{2.17}$$

Noting the form of the synthesis window and the final constraint in Equation (2.16), the restriction that $b[n]$ be strictly nonzero can be relaxed slightly: $b[n]$ can be zero where $w_{\mathrm{PR}}[n]$ is also zero; if the synthesis window $v[n]$ is defined to be zero at those points, the perfect reconstruction condition is met. This latter design method will come into play in the frequency-domain sinusoidal synthesizer to be discussed in Section 2.5.

### The STFT as a heterodyne filter bank

In [111, 115, 116, 20], where the STFT is defined as in Equation (2.5) and the expansion functions have an absolute time reference, the transform can be interpreted as a filter bank with a heterodyne structure. Starting with Equation (2.5),

$$\tilde{X}[k,n] \;=\; \sum_{m=n}^{n+N-1} \tilde{w}[n-m]x[m]e^{-j\omega_k m}, \tag{2.18}$$

the substitution

$$x_k[m] \;=\; x[m]e^{-j\omega_k m} \tag{2.19}$$

yields an expression that is immediately recognizable as a convolution:

$$\tilde{X}[k,n] \;=\; \sum_{m=n}^{n+N-1} \tilde{w}[n-m]x_k[m]. \tag{2.20}$$

The filter $\tilde{w}[n]$ is typically lowpass; it thus extracts the baseband spectrum of $x_k[m]$. According to the modulation relationship defined in Equation (2.19), $x_k[m]$ is a version of $x[m]$ that has been modulated down by $\omega_k$; thus, the baseband spectrum of $x_k[m]$ corresponds to the spectrum of $x[m]$ in the neighborhood of frequency $\omega_k$. In this way, the $k$-th branch of the STFT filter bank extracts information about the signal in a frequency band around $\omega_k = 2\pi k/K$.

In the time domain, $\tilde{X}[k,n]$ can be interpreted as the amplitude envelope of a sinusoid with frequency $\omega_k$. This perspective leads to the framework for signal reconstruction based on the filter bank interpretation of the STFT; this framework is known as the *filter bank summation* (FBS) method. The idea is straightforward: the signal can be reconstructed by modulating each of these envelopes to the appropriate frequency and summing the resulting signals. This construction is given by

$$\hat{x}[n] \;=\; \sum_{k} \tilde{X}[k,n]e^{j\omega_k n}, \tag{2.21}$$

which can be manipulated to yield perfect reconstruction conditions [1, 111]; this non-subsampled case is not very general, however, so these constraints will not be derived here. Rather, Equation (2.21) is given to indicate the similarity of the STFT signal model and the sinusoidal model. Each of the components in the sum of Equation (2.21) can be likened to a partial; the STFT $\tilde{X}[k,n]$ is then the time-varying amplitude of the $k$-th partial. Note that in the phase-localized STFT formulated in Equation (2.3), the corresponding reconstruction formula is

$$\hat{x}[n] \;=\; \sum_{k} X[k,n], \tag{2.22}$$

where the STFT $X[k,n]$ corresponds to a partial at frequency $\omega_k$ rather than its amplitude envelope.

Figure 2.2 depicts one branch of a heterodyne STFT filter bank and provides an equivalent structure based on modulated filters [20]. Mathematically, the equivalence is

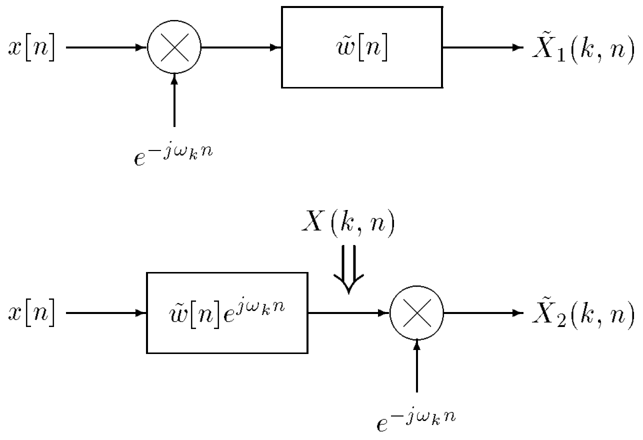

FIGURE 2.2: One channel of a heterodyne filter bank for evaluating the STFT $\tilde{X}[k,n]$ defined in Equation (2.5). The two structures are equivalent as indicated in Equation (2.23). The STFT $X[k,n]$ as defined in Equation (2.3) is an intermediate signal in the second structure.

straightforward:

$$\begin{aligned} \tilde{X}_1[k,n] &= \sum_m \tilde{w}[n-m]\left(x[m]e^{-j\omega_k m}\right) \\ &= e^{-j\omega_k n}\sum_m \left(\tilde{w}[n-m]e^{j\omega_k(n-m)}\right)x[m] \;=\; \tilde{X}_2[k,n]. \end{aligned} \tag{2.23}$$

Given the relationship in Equation (2.8), namely that $X[k,n] = \tilde{X}[k,n]e^{j\omega_k n}$, it is clear that $X[k,n]$ is the immediate output of the modulated filter $\tilde{w}[n]e^{j\omega_k n}$ without the ensuing modulation to baseband. This observation, which is indicated in Figure 2.2, serves as motivation for interpreting the STFT of Equation (2.3) as a modulated filter bank.

### The STFT as a modulated filter bank

Modulated filter banks, in which the filters are modulated versions of a prototype lowpass filter, have been of considerable interest in the recent literature [20, 2, 4]. In part, this interest has stemmed from the realization that the STFT can be implemented with a modulated filter bank structure. Indeed, the STFT of Equation (2.3) corresponds exactly to a modulated filter bank of the general form shown in Figure 2.3. This filter bank is markedly different from the heterodyne structure in that the subband signals are not amplitude envelopes but are actual signal components that can be likened to partials, which will prove conceptually useful in extending the STFT to the general sinusoidal model.

The modulated filter bank of Figure 2.3 implements an STFT analysis-synthesis

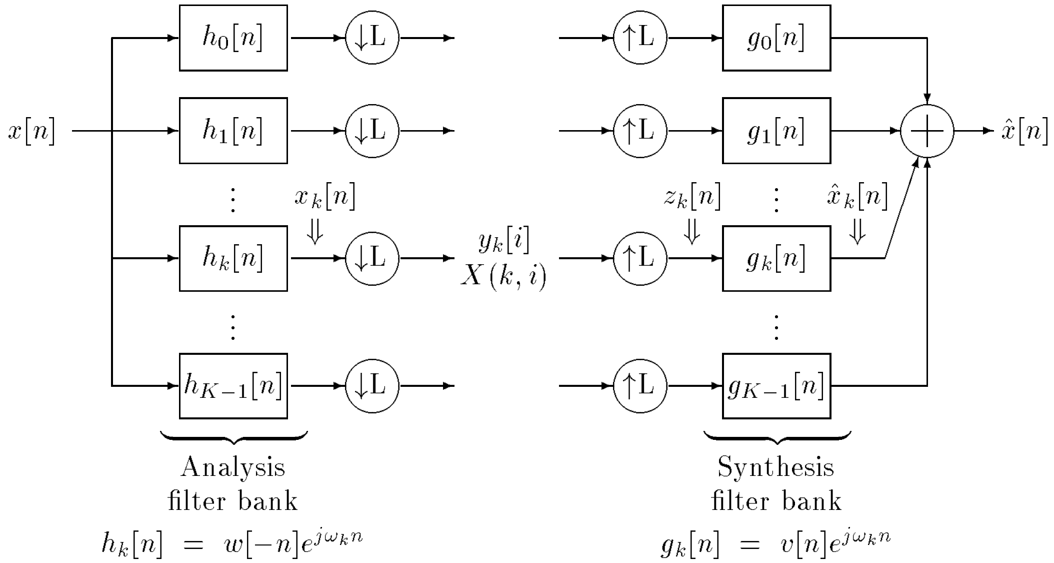

FIGURE 2.3: Interpretation of the short-time Fourier transform as a modulated filter bank. The subband signals are labeled to match the formulation in the text.

if the filters are defined as

$$h_k[n] = w[-n]e^{j\omega_k n} \tag{2.24}$$

$$g_k[n] = v[n]e^{j\omega_k n}. \tag{2.25}$$

Note the time-reversal of the window $w[n]$ in the definition of the analysis filter $h_k[n]$; the time-reversal appears here because the window in Equation (2.3) is not thought of in a time-reversed fashion as in Equation (2.5). Using the notation in Figure 2.3, the subband signals in the STFT filter bank are given by

$$\begin{aligned} x_k[n] &= \sum_m h_k[m]x[n-m] && (2.26)\\ &= \sum_m w[-m]x[n-m]e^{j\omega_k m} && (2.27)\\ &= \sum_m w[m]x[n+m]e^{-j\omega_k m} && (2.28)\\ &= X[k,n] && (2.29)\end{aligned}$$

$$\begin{aligned} y_k[i] &= x_k[iL] && (2.30)\\ &= X(k,i) && (2.31)\end{aligned}$$

$$z_k[n] = x_k[n]\sum_i \delta[n-iL], \tag{2.32}$$

where the last expression simply describes the effect of successive downsampling and upsampling on the signal $x_k[n]$. Again, note that the subband signals are essentially the partials of the signal model, and are not amplitude envelopes as in the heterodyne structure of the traditional STFT filter bank.

In the framework of [111], namely the STFT as given in Equation (2.5), the overlap-add and filter bank summation synthesis methods lead to different perfect reconstruction constraints which can be interpreted as duals. For the phase-localized definition of the STFT, on the other hand, the overlap-add and filter bank methods lead directly to the same constraint:

$$\begin{aligned}
\hat{x}[n] &= \sum_{k=0}^{K-1} \hat{x}_k[n] && (2.33)\\
&= \sum_k (z_k[n] * g_k[n]) && (2.34)\\
&= \sum_k \sum_l g_k[l] z_k[n-l] && (2.35)\\
&= \sum_k \sum_l v[l] e^{j\omega_k l} x_k[n-l] \sum_i \delta[n-l-iL] && (2.36)\\
&= \sum_k \sum_l \sum_i \sum_m v[l] w[m] x[n-l+m] e^{j\omega_k (l-m)} \delta[n-l-iL]. && (2.37)
\end{aligned}$$

For $\omega_k = 2\pi k/K$, the summation over the frequency index $k$ can be expressed as

$$\sum_{k=0}^{K-1} e^{j\omega_k (l-m)} = K \sum_r \delta[l-m+rK]. \tag{2.38}$$

If $|l-m| < K$ for all possible combinations of $l$ and $m$, then the only relevant term in the right-hand sum is for $r=0$, in which case the equation simplifies to

$$\sum_{k=0}^{K-1} e^{j\omega_k (l-m)} = K\delta[l-m]. \tag{2.39}$$

The restriction on the values of $l$ and $m$ corresponds to the constraint $K \geq N$ discussed in the treatment of overlap-add synthesis; namely, time-domain aliasing is introduced if $l$ and $m$ do not meet this criterion. Further consideration of time-domain aliasing is deferred until Section 2.2.2.

As in the discussion of OLA synthesis, it is assumed at this point that time-domain aliasing is not introduced. Then, the FBS reconstruction formula can be rewritten as

$$\begin{aligned}
\hat{x}[n] &= K \sum_l \sum_i v[l] \delta[n-l-iL] \sum_m w[m] x[n-l+m] \delta[l-m] && (2.40)\\
&= K x[n] \sum_l \sum_i w[l] v[l] \delta[n-l-iL] && (2.41)\\
&= K x[n] \sum_i w[n-iL] v[n-iL]. && (2.42)
\end{aligned}$$

The design constraint for perfect reconstruction, within a gain term, is then exactly the same as in the overlap-add synthesis approach:

$$\sum_i w[n-iL]v[n-iL] \; = \; 1. \tag{2.43}$$

Because of this equivalence, the analysis-synthesis window pairs described earlier can be used as prototype functions for perfect reconstruction modulated filter banks.

Note that if $L > 1$, the synthesis filter bank interpolates the subband signals. In the nonsubsampled case $L = 1$, when no interpolation is needed, perfect reconstruction can be achieved with any analysis-synthesis window pair for which $\sum_n w[n]v[n] \neq 0$. For example, the synthesis can be performed with the trivial filter bank $g_k[n] = \delta[n]$ if the analysis window satisfies the constraint

$$\sum_i w[n-i] \; = \; 1, \tag{2.44}$$

which indeed holds for *any* window, within a gain term. The generality of this constraint is an example of the design flexibility that results from using oversampled or overcomplete approaches [70, 64, 125]. Reiterating the modeling implications, the STFT signal model is

$$x[n] \; = \; \sum_k X[k,n] \; = \; \sum_k \tilde{X}[k,n]e^{j\omega_k n}. \tag{2.45}$$

In the modulated filter bank case, the subband signals can be viewed as the partials of the sinusoidal model; in the heterodyne case, the subband signals are instead lowpass amplitude envelopes of the partials. Furthermore, the phase of $X[k,n]$ is the phase of the $k$-th partial whereas the phase of $\tilde{X}[k,n]$ is the phase of the envelope of the $k$-th partial; the former phase measurement is needed for the sinusoidal model. In the next section, it will be shown that rigid association of the subband signals to partials is basically inappropriate for either case; the modulated STFT analysis filter bank, however, more readily provides the information necessary to derive a generalized sinusoidal signal model.

### 2.2.2 Limitations of the STFT and Parametric Extensions

The interpretation of the STFT as a modulated filter bank leads to a variety of modeling implications. These issues in some sense revolve around the nonparametric representation of the signal in terms of subbands and the use of a rigid filter bank for synthesis. This section deals with the limitations of the STFT; the considerations motivate parametric extensions of the STFT that overcome some of these limitations.

#### Partial tracking

The most immediate limitation of the short-time Fourier transform results from its fixed structure. A sinusoid with time-varying frequency will move across bands; this

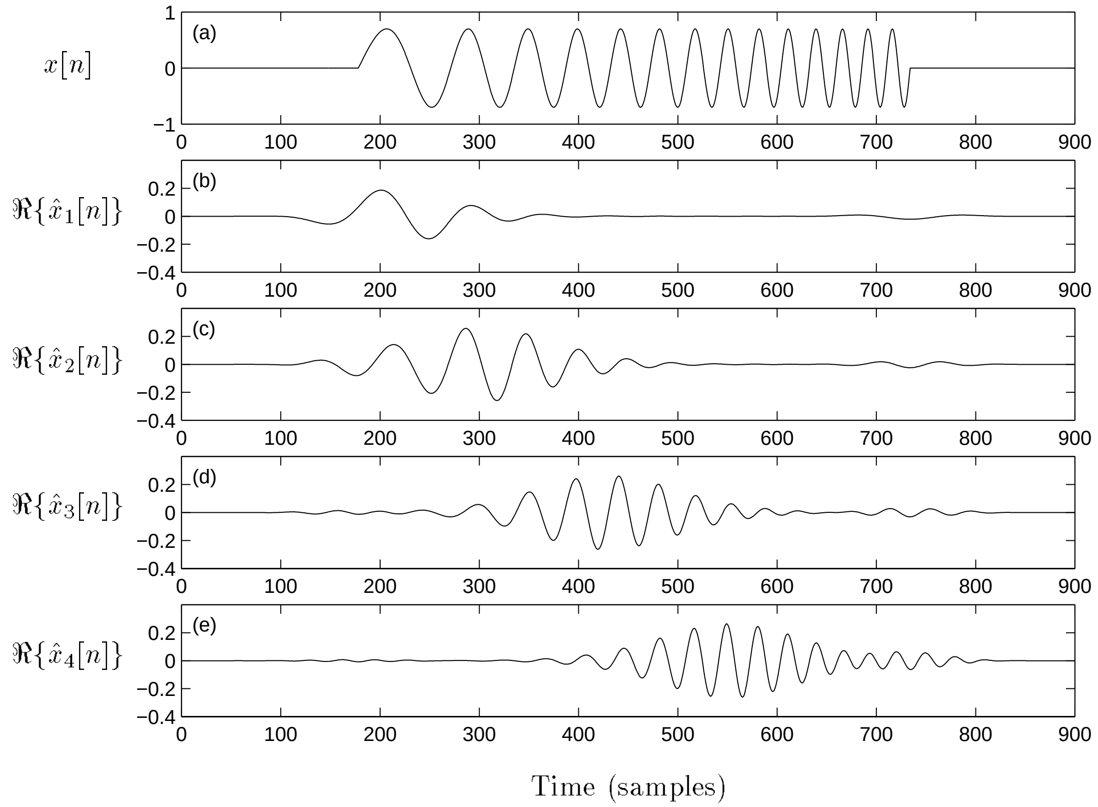

FIGURE 2.4: Reconstructed subband signals in an nonsubsampled STFT filter bank model of a chirp signal. The signals $\hat{x}_k[n]$ correspond to those labeled in Figure 2.3 for $k = \{1, 2, 3, 4\}$. In the simulation, $N = 128$, $K = 128$, and $L = 1$; $w[n]$ and $v[n]$ are square-root Hanning windows.

evolution leads to delocalization of the representation and a noncompact model. Consider the example shown in Figure 2.4, in which a sinusoid of linearly increasing frequency, *i.e.* a linear *chirp*, is modeled by a nonsubsampled STFT filter bank where the analysis and synthesis filter prototypes are both square-root Hanning windows ($c = \frac{1}{2}$). The parameters of the STFT are $K = 128$, $L = 1$, and $N = 64$; the chirp frequency starts at $\omega_0 = 2\pi/K$ and increases by that amount every 250 samples.

Figure 2.4 shows the real parts of the reconstructed subband signals for bands $k = 1, 2, 3, 4$. It is necessary to consider the real parts for the following reason: the subband signals in the STFT are complex-valued as a result of the complex modulation of the filters. For real signals, the STFT yields a conjugate symmetric representation like the underlying DFT; each of these subband signals has a conjugate version. This observation motivates cosine-modulated filter banks where the prototype filters are modulated with a real cosine instead of a complex sinusoid. Then, the subband signals are real-valued, which is certainly desirable in some cases; here, however, it is problematic since the phase provided by the complex filter bank is important for sinusoidal modeling as will be seen. While cosine-modulated filter banks have interesting and significant properties [2, 20, 4, 12, 59], they are an offshoot of the progression of ideas that leads to the sinusoidal model and will not be considered in depth here because of this phase problem.

Returning to the example of Figure 2.4, it is clear that the subbands of the fixed filter bank do not provide a compact representation of the chirp signal. As the chirp evolves in time, it moves across the bands of the filter bank, and as a result the STFT does not identify this as a single evolving sinusoid but instead as a conglomeration of short-lived components, *i.e.* the subband signals shown in Figure 2.4. Whereas this may seem useful in that it carries out a granulation of the chirp signal (Section 1.5.4), inspection of the signal components show that the subband grains are not well-localized in time; note that the transients in the original signal are manifested in all of the subband signals as pre-echoes. Figure 2.5 shows the model of the same chirp signal using a subsampled STFT filter bank with $L = 64$. This example is perhaps more practical than the nonsubsampled case in that there is much less data in the representation, but this practicality comes at the cost of more substantial localization problems in the subbands. Perfect reconstruction can be achieved in this case; the various artifacts cancel in the synthesis. The signal decomposition, however, is virtually useless for modifications because of these delocalization artifacts; if the subbands are modified, *e.g.* quantized, the subband artifacts will not be properly cancelled and will lead to artifacts in the final synthesis.

In pseudo-periodic musical signals, the frequencies of the harmonics vary as the pitch evolves in time; it is intuitively desirable that the sum-of-partials model in such cases should be an aggregation of chirps whose frequencies are coupled while changing in time in a complex way. In this case, unlike the single chirp case, all of the STFT filter bank subbands will generally have significant energy throughout the duration of the signal, so inspection of the subbands will not necessarily indicate that the various partials are moving *across* the bands. When all of the subbands have significant energy, it may seem reasonable to interpret the subbands as the partials of the sinusoidal model as has been discussed; this perspective, however, is in contention with the physical foundation of the natural signal. The generating mechanism for a signal whose harmonic structure varies in time is a system with a physical parameter, such as a string length, that is correspondingly time-varying, and a meaningful representation should capture this foundation. Rather than imposing structure on the partials by restricting them to exist within subbands as in the STFT model, the time-frequency evolution of the partials should instead be tracked. As will be seen, this tracking effort is what makes the sinusoidal model fundamentally signal-adaptive.

One approach to the problem of partial tracking in an STFT filter bank is to make the filter bank pitch-adaptive so that the subbands do correspond to physically reasonable partials; in that method, which was considered in a preliminary fashion in

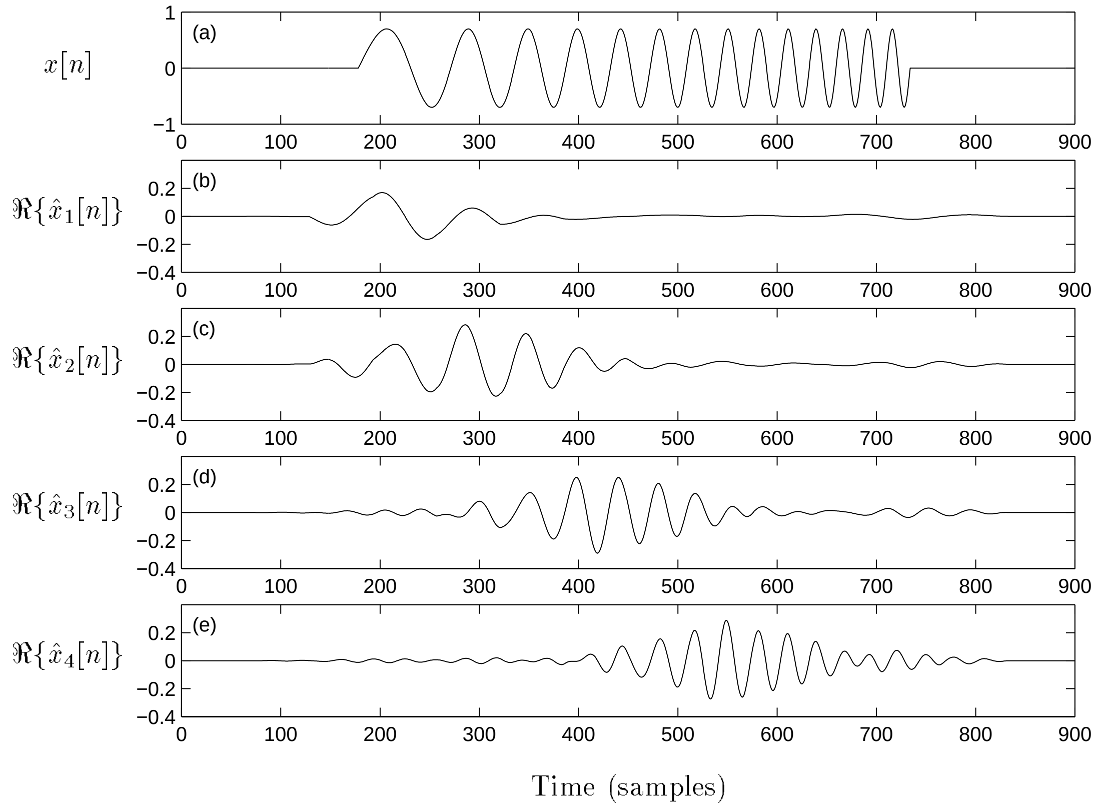

FIGURE 2.5: Reconstructed subband signals in a subsampled STFT filter bank model of a chirp signal. The signals $\hat{x}_k[n]$ correspond to those labeled in Figure 2.3 for $k = \{1, 2, 3, 4\}$. In the simulation, $N = 128$, $K = 128$, and $L = 64$; $w[n]$ and $v[n]$ are square-root Hanning windows.

[126], signal adaptivity improves the model. A pitch-adaptive filter bank, however, does not account for the more general case of signals composed of nonharmonic partials with unrelated frequency evolution behavior, for instance a percussive sound such as a cymbal clash. The intent of modeling arbitrary signals necessitates using a more general model.

### Time-domain aliasing cancellation

Time-domain aliasing was mentioned in the discussions of both the overlap-add and the filter bank summation synthesis methods; in those treatments, it was assumed that $K$ was large enough that time-domain aliasing was not introduced. In this section, the issue of time-domain aliasing is explored; the treatment leads to general perfect reconstruction constraints for modulated filter banks and various implications for signal modeling. This issue is discussed here more for the sake of completeness than as a prerequisite for the development of the general sinusoidal model. Essentially, time-domain aliasing cancellation is a fix that allows for perfect reconstruction despite a lack in frequency resolution; with this in mind, the importance of frequency resolution in sinusoidal modeling implies that STFT filter banks that incorporate time-domain aliasing cancellation will not be of interest in future considerations.

For a signal $a[n]$ of length $N$ on $[0, N-1]$, application of a size $K$ DFT followed by a size $K$ IDFT corresponds to

$$\begin{aligned} \hat{a}[n] &= \frac{1}{K}\sum_{k=0}^{K-1}\left\{\sum_{m=0}^{N-1} a[m]e^{-j2\pi km/K}\right\}e^{j2\pi kn/K} && (2.46)\\ &= \frac{1}{K}\sum_{m=0}^{N-1} a[m]\sum_{k=0}^{K-1} e^{j2\pi k(n-m)/K}. && (2.47)\end{aligned}$$

Using the simplification for the sum over $k$ given in Equation (2.38) yields

$$\begin{aligned} \hat{a}[n] &= \sum_{m=0}^{N-1} a[m]\sum_{r=-\infty}^{\infty}\delta[n-m+rK] && (2.48)\\ &= \sum_{r} a[n+rK], && (2.49)\end{aligned}$$

where the $r$ values in the sum of the last expression correspond to values of $n + rK$ that fall within the span of the signal, namely

$$0 \leq n+rK \leq N-1 \quad \text{for} \quad 0 \leq n \leq N-1. \qquad (2.50)$$

This formulation explains the condition on $K$ imposed in the earlier treatments; if $K \geq N$, time-domain aliasing is not introduced because only the $r = 0$ term contributes to the reconstruction. On the other hand, if $K < N$, the signal is aliased in the time domain. Fundamentally, this aliasing is a result of insufficient spectral sampling of the continuous function $A(e^{j\omega})$, the discrete-time Fourier transform (DTFT) of $a[n]$, and is thus analogous to the frequency-domain aliasing that occurs when a continuous time-domain signal is sampled below the Nyquist rate. The DTFT, the DFT, and spectral sampling are discussed further in Section 2.5.1.

The effect of time-domain aliasing on the perfect reconstruction condition can be readily formalized; the following derivation uses the overlap-add synthesis framework, but the same condition results in the filter bank summation approach, within a gain factor of $K$. For the signal segment

$$a_i[n] \;=\; w[n-iL]x[n], \tag{2.51}$$

the reconstructed version of the segment is given by

$$\hat{a}_i[n] \;=\; \sum_r a_i[n+rK] \;=\; \sum_r w[n+rK-iL]x[n+rK]. \tag{2.52}$$

The OLA synthesis of the signal, with synthesis window $v[n]$, is given by

$$\hat{x}[n] \;=\; \sum_i v[n-iL]\hat{a}_i[n]. \tag{2.53}$$

Substituting for $\hat{a}_i[n]$ and changing the order of the sums yields

$$\hat{x}[n] \;=\; \sum_r x[n+rK]\sum_i v[n-iL]w[n+rK-iL]. \tag{2.54}$$

If $\hat{x}[n] = x[n]$ is to hold, every term but $r = 0$ must be cancelled in the other sum; the perfect reconstruction constraint is thus

$$\sum_i v[n-iL]w[n+rK-iL] \;=\; \delta[r]. \tag{2.55}$$

In the nonsubsampled case with $v[n] = \delta[n]$, this simplifies to

$$w[rK] \;=\; \delta[r], \tag{2.56}$$

which is reminiscent of the constraint for designing interpolation filters [116, 127]. Note that since the time index is the start of the window in this treatment, the most appropriate synthesis window is actually given by $v[n] = \delta[n-n_0]$, where $n_0$ corresponds to the middle of the analysis window. The final constraint on the analysis window is then $w[n_0 + rK] = \delta[r]$, which is satisfied by any function with zeros at $n_0 + rK$ for all $r \neq 0$ and a nonzero value at $n_0$, which can be scaled to unity for gain compensation. A useful class of windows that meet this constraint can be constructed by multiplying a perfect reconstruction window by an appropriate sinc function. As mentioned earlier, perfect reconstruction windows can be virtually arbitrary in the nonsubsampled case; here, the formulation is most appealing if the perfect reconstruction windows under consideration are those that apply to subsampled cases. This class of aliasing cancellation windows are given by

$$w[n] \;=\; w_{\mathrm{PR}}[n]\frac{\sin\left[\pi(n-n_0)/K\right]}{\pi(n-n_0)}, \tag{2.57}$$

where the sinc function, as written, will introduce a gain of $1/K$. The frequency response of the resultant window is the spectrum of $w_{\mathrm{PR}}[n]$ convolved with an ideal lowpass filter with cutoff frequency $\pi/K$; this convolution relationship implies that $w[n]$ is a broader lowpass filter than $w_{\mathrm{PR}}[n]$, which corroborates the previous statement that time-domain aliasing cancellation and lack of frequency resolution are coupled.

As indicated above, the design of time-domain aliasing cancellation windows in the subsampled case is more restricted than in the nonsubsampled case; in other words, there is limited freedom in the design of subsampled STFT filter banks that employ time-domain aliasing cancellation. The subsampling limits the design possibilities since it introduces frequency-domain aliasing, the cancellation of which is an underlying principle in the equivalent constraints of Equations (2.12) and (2.43), and is indeed part of the general constraint given above in Equation (2.55). The critically sampled case $L = K$ is of special interest since the representation and the original signal intrinsically contain the same amount of data. For critical sampling, however, it can be shown that the only FIR solutions correspond to windows with $N = K$ nonzero coefficients [2, 120]. In the straightforward solution of this form, the $N$ nonzero coefficients are all in the interval $[0, N-1]$. Intuitively, there are no solutions of this form for $N < K$ since gaps would result in the window overlap and various regions of the signal would simply be missed in the analysis-synthesis. On the other hand, the reason that there are no solutions for $N > K$ is less intuitive; this result is proved in [2, 120]. In the critically sampled case, then, the STFT in effect implements a block transform with block size $N$; quantization then leads to discontinuities at the block boundaries, which results in undesirable frame rate artifacts in audio and blockiness in images. Furthermore, pre-echo distortion occurs in the reconstruction where the original signal has transient behavior; pre-echo is a common problem in near-perfect reconstruction signal models such as filter banks with subband quantization [7].

The requirement that $N = K = L$ in the critically sampled case means that there are no critically sampled perfect reconstruction STFT filter banks that employ time-domain aliasing cancellation. However, time-domain aliasing cancellation can be incorporated in critically sampled cosine-modulated filter banks; such filter banks are commonly used in audio coding [12, 7, 9, 16, 17]. The ability to use time-domain aliasing cancellation in a cosine-modulated filter bank is connected to the result that the expansion functions in a cosine-modulated filter bank can have good time and frequency localization [2]. Note that the lapped orthogonal transforms (LOT) mentioned in Section 1.4 belong to this class of filters. In the LOT, the representation is critically sampled but all of the basis functions are smooth and extend beyond the boundaries of the signal segment or block; this overlap reduces the artifacts caused by quantization. Quantization effects can also be reduced by oversampling; one advantage of overcomplete representations is that they exhibit a robustness to quantization noise that is proportional to the redundancy of the representation [70, 64, 92, 125].

### Time-frequency localization

As discussed above, the design of STFT filter banks is extremely limited in the critically sampled case. The only real-valued prototype windows that lead to orthogonal perfect reconstruction filter banks are rectangular windows [2]. This result is a discrete-time equivalent of the Balian-Low Theorem, which states that there are no continuous-time orthogonal short-time Fourier transform bases that are localized in time and frequency, where the localization is measured in terms of $\Delta_t$ and $\Delta_\omega$ from Equations (1.30) and (1.31); either or both of these uncertainty widths are unbounded for orthonormal STFT bases. This problem motivates the use of cosine-modulated filter banks, which can achieve good time-frequency localization [2].

Further issues regarding time-frequency localization and filter banks are beyond the scope of this thesis; this issue will thus not be addressed further, with the exception of various considerations of signal expansions, which have a fundamental relationship to filter banks. The point of this discussion is simply to cite the result that there are some difficulties with critically sampled STFT filter banks, and that oversampling is thus required in order for STFT filter banks to perform well. The use of oversampling, however, is contrary to the goal of data reduction. This problem is solved in the sinusoidal model by applying a parametric representation to the STFT to achieve compaction.

### Modifications of the STFT

Various signal modifications based on the STFT have been discussed in the literature [1, 111, 114, 115, 112, 117, 128, 129]. In approaches where the modifications are based directly on the function $X(k, i)$, the techniques are inherently restricted to a rigid framework because the signal is being modeled in terms of subbands which interact in complicated ways in the reconstruction process. The restrictive framework is exactly this: a modification is carried out on the subband signals and the effect of the modification on the output signal is then formulated [111, 115]. This approach is much different from the desired framework of simply carrying out a particular modification on the original signal.

In some approaches, modifications are based on the STFT magnitude only; the magnitude is first modified and then a phase that will minimize synthesis discontinuities is derived [117, 128, 129]. This removal of the phase essentially results in a parametric representation that is more flexible than the complex subband signals. It is important to note that this magnitude-only description has the same caveat as other parametric models: for the case of no modification, the magnitude-only description is not capable of perfect reconstruction.

In the critically sampled case, there is a one to one correspondence between signals and short-time Fourier transforms; because it is equivalently a basis expansion, there is no ambiguity in the relationship between the domains. In the oversampled case, however, many different STFTs will yield the same signal. This multiplicity is obviated by considering the simplest case: $L = 1$ and $v[n] = \delta[n]$; the analysis window $w[n]$, which derives the STFT, is virtually unrestricted. Such an overcomplete representation has a higher dimension than the signal space, meaning that some modifications in that space may have no effect on the signal or may produce an otherwise unexpected result; in deriving a phase for the STFT magnitude for synthesis in the overcomplete case, there are thus *consistency* or *validity* concerns that arise [130].

The issues of aliasing cancellation and validity, among others, indicate the fundamental point: the synthesis model limits the modification capability. Given that the most effective modification methods for the STFT rely on parameterizations of the STFT, there is in some sense no need to use a rigid filter-based structure for synthesis. This observation is the fundamental motivation for the sinusoidal model, which relies on an STFT analysis filter bank for parameter estimation, but thereafter utilizes a fully parametric synthesis to circumvent issues such as frame boundary discontinuities, consistency, and aliasing cancellation. The channel vocoder and the phase vocoder are the two fundamental steps in the progression from the STFT to the sinusoidal model.

### The channel vocoder

The term *vocoder*, a contraction of *voice* and *coder*, was coined to describe an early speech analysis-synthesis algorithm [131]. In particular, the channel vocoder originated as a voice coder which represented a speech signal based on the characteristics of the STFT filter bank channels or subbands. Specifically, the speech is filtered into a large number of channels using an STFT analysis filter bank. Each of the subbands is modeled in terms of its short-time energy; with respect to the $k$-th channel, this provides an amplitude envelope $A_k[n]$ which modulates a sinusoidal oscillator at the channel center frequency $\omega_k$. The outputs of these oscillators are then accumulated to reconstruct the signal. Note that the term “vocoder” has at this point become a general designation for a large number of algorithms which are by no means limited to voice coding applications.

### The phase vocoder

The channel vocoder parameterizes the subband signal in terms of its energy or amplitude only; the phase vocoder is an extension that includes the phase behavior in

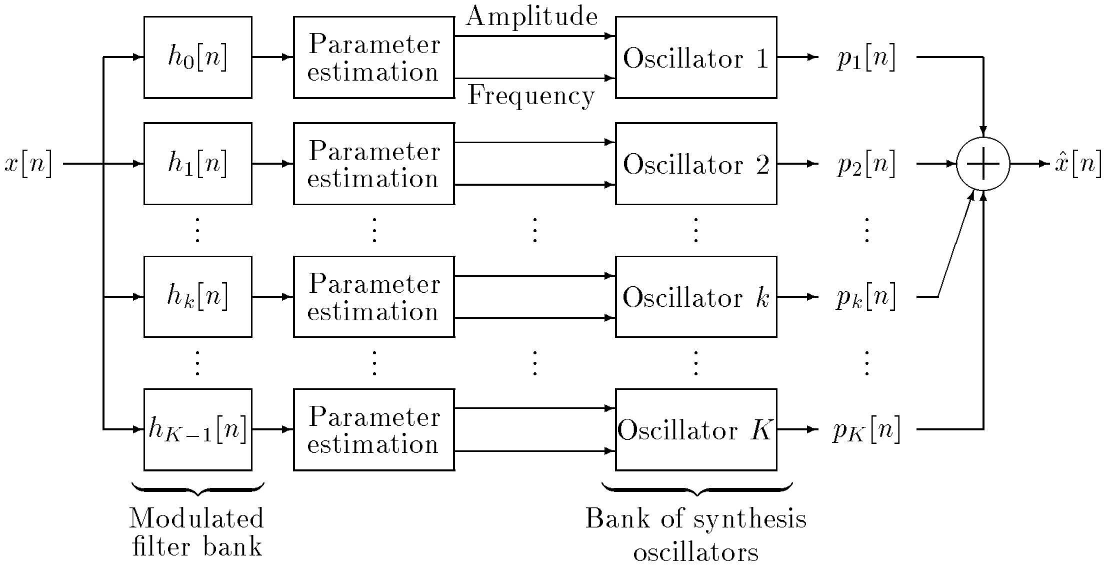

FIGURE 2.6: Block diagram of the phase vocoder. The amplitude and frequency (total phase) control functions for the $K$ oscillators are derived from the filter bank output signals by the parameter estimation blocks.

the model parameterization as well. There are a number of variations, but in general the term refers to a structure like the one shown in Figure 2.6, where the subband signals are parameterized in terms of magnitude envelopes and functions that describe the frequency and phase evolution; these serve as inputs to a bank of oscillators that reconstruct the signal from the parametric model [113, 118, 116, 119]. This approach has been widely applied to modification of speech signals; the success of such approaches substantiates the previous contention that modifications are enabled by the incorporation of a parametric model and a parametric synthesis. Note that if the analysis filter bank is subsampled, the sample-rate oscillator control functions are derived from the subsampled frame-rate STFT representation.

### General sinusoidal models

The phase vocoder as depicted in Figure 2.6 does not solve the partial tracking problem discussed earlier; while its parametric nature does enable modifications, it is still of limited use for modeling evolving signals. A further generalization leads to the sinusoidal model. The fundamental observation in the development of the sinusoidal model is that if the signal consists of one nonstationary sinusoid such as a chirp, then synthesis can be achieved with one oscillator. There is no need to implement an oscillator for every branch of the analysis filter bank. Instead, the outputs of the analysis bank can be examined across frequency for peaks, which correspond to sinusoids in the signal. These spectral

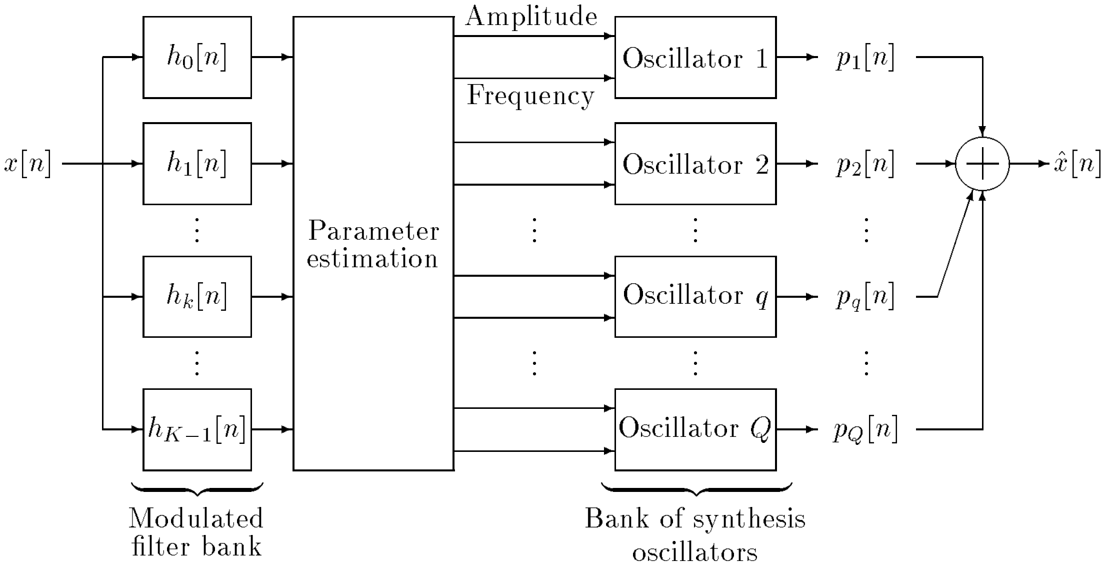

FIGURE 2.7: Block diagram of the general sinusoidal model. The amplitude and frequency (total phase) control functions are derived from the filter bank outputs by tracking spectral peaks in time as they move from band to band for an evolving signal. The parameter estimation block detects and tracks spectral peaks; unless $Q$ is externally constrained, the number of peaks detected dictates the number of oscillators used for synthesis.

peaks can then be tracked from frame to frame as the signal evolves, and only one oscillator per tracked peak is required for synthesis. This structure is depicted in Figure 2.7.

For the chirp signal used in Figures 2.4 and 2.5, a sinusoidal model with one oscillator yields the reconstruction shown in Figure 2.8(b). The model data for the reconstruction in Figure 2.8(b) is extracted from the same STFT produced by the subsampled analysis filter bank of the Figure 2.5 example. With respect to data reduction, the one-partial sinusoidal model in Figure 2.8 is basically characterized by three real numbers $\{A, \omega, \phi\}$ for each signal frame; for real signals, the STFT filter bank model consists of $K/2$ complex numbers for each frame, so the compression achieved is significant; this is of course less drastic for complicated signals with many partials. Note that this compression is accompanied by the inability to carry out perfect reconstruction. A primary reconstruction inaccuracy or artifact in the sinusoidal model is pre-echo, which is evident in Figure 2.8. This problem is discussed further in Section 2.6; in Chapter 3, methods for alleviating the pre-echo distortion are developed. Note also that the sinusoidal model provides a better description of the signal behavior than the filter bank decomposition; this example illustrates how a compact parametric model is useful for analysis.

In the general sinusoidal model, there are no strict limitations on $N$, $K$, and $L$ for the analysis filter bank. Typically, $K > N$, meaning that oversampling in frequency is

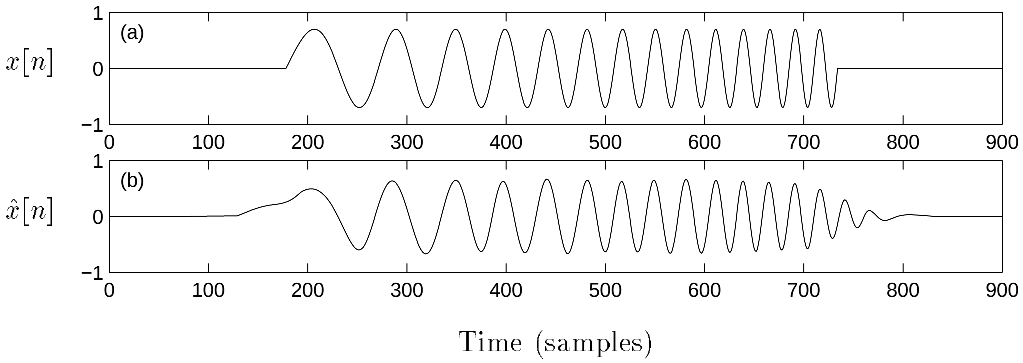

FIGURE 2.8: One-component sinusoidal model of the chirp signal from Figure 2.4 using the same analysis filter bank as in that example.

used, which in some cases yields a more accurate model than critical sampling ($K = N$) as will be seen in the next section. Note that an increase in $K$ corresponds to adding more channels to the filter bank and decreasing the frequency spacing between channels; because each filter is simply a modulated version of the prototype window, however, the resolution of the individual channel filters is not affected by a change in $K$. Also, it is common to use a hop size of $L = N/2$ to achieve data reduction. Of course, gaps result in the analysis if $L > N$ as in the filter bank case, but in the sinusoidal model such gaps can be filled in the reconstruction via parameter interpolation.

## 2.3 Sinusoidal Analysis

The analysis for the sinusoidal model is responsible for deriving a set of time-varying model parameters, namely the number of partials $Q[n]$, which may be constrained by rate or synthesis computation limits [132], and the partial amplitudes $\{A_q[n]\}$ and total phases $\{\Theta_q[n]\}$. As mentioned, these parameters are assumed to be slowly varying with respect to the sample rate, so the estimation process can be reliably carried out at a subsampled rate. In [57, 36], this analysis is done using a short-time Fourier transform followed by spectral peak picking; this procedure was conceptually motivated in the preceding discussion of the STFT. The following sections examine this analysis method in detail; alternative approaches are also discussed.

### 2.3.1 Spectral Peak Picking

The analysis for the sinusoidal model is similar to many scenarios in which the sinusoidal content of a signal is of interest. Approaches based on Fourier transforms have been traditionally applied to these problems. In such methods, the signal is transformed into the Fourier domain and the peaks in the spectral representation are interpreted as sinusoids. In this section, the use of the discrete Fourier transform in this framework is considered; various resolution limits are demonstrated. The relationship of the discrete-frequency DFT to the continuous DTFT underlies some of the issues here; a discussion of this relationship, however, is deferred to Section 2.5.1.

### A single sinusoid

The case of identifying a single time-limited complex sinusoid is of preliminary importance for these considerations. For the signal

$$x[n] \;=\; \alpha_0 e^{j\omega_0 n} \tag{2.58}$$

defined on the interval $n \in [0, N-1]$, where $\alpha_0$ is a complex number that entails the magnitude and phase of the sinusoid, a DFT of size $N$ is given by

$$X_N[k] \;=\; \alpha_0\; e^{j\omega_0\left(\frac{N-1}{2}\right)}\; e^{-j\pi k\left(\frac{N-1}{N}\right)} \left[\frac{\sin\left(N\left[\frac{\pi k}{N} - \frac{\omega_0}{2}\right]\right)}{\sin\left(\frac{\pi k}{N} - \frac{\omega_0}{2}\right)}\right], \tag{2.59}$$

where the subscript $N$ denotes the size of the DFT. This treatment will focus on the estimation of sinusoids based on peaks in the magnitude of the DFT spectrum, so the ratio of sines in the above expression is of more importance than the preceding linear phase term. If the frequency of the sinusoid can be expressed as

$$\omega_0 \;=\; \frac{2\pi k_0}{N}, \tag{2.60}$$

namely if it is equal to a *bin* frequency of the DFT, the numerator in this ratio is zero-valued for all $k$, meaning that the DFT itself is zero-valued everywhere except at $k = k_0$, where the denominator of the ratio is zero. For $k = k_0$, the ratio takes on a value $N$ by L'Hôpital's rule, so the DFT magnitude is $N|\alpha_0|$; the phase at $k = k_0$ is given simply by $\arg \alpha_0$. Thus, when $\omega_0$ corresponds to a bin frequency, the sinusoid can be perfectly identified as a peak in the DFT magnitude spectrum, and its magnitude and phase can be extracted from the DFT. For sinusoids at other frequencies, however, the $N$-point DFT has a less simple structure. In this case, the signal is indeed represented exactly because the DFT is a basis expansion; however, in terms of spectral peak picking it is erroneous to interpret the peak in such a DFT as a sinusoid in the signal. These cases are depicted in Figures 2.9(a) and 2.9(b), respectively.

### Oversampling and frequency resolution

For the case of the off-bin frequency illustrated in Figure 2.9(b), the sinusoid cannot be immediately identified in the DFT spectrum, and the DFT representation of

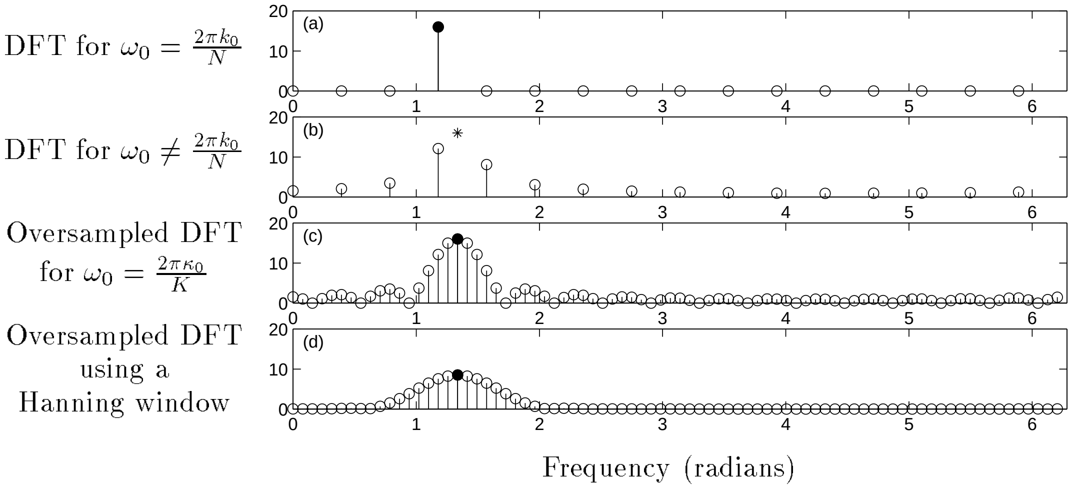

Figure 2.9: Estimation of a single sinusoid with the DFT. In (a), the sinusoid is at the bin frequency $2\pi k_0/N$ for $N = 16$ and $k_0 = 3$, so an $N$-point DFT identifies the sinusoid exactly. In (b), the frequency is $2\pi(k_0 + 0.4)/N$ as indicated by the asterisk in the plot; the sinusoid is not identified by the DFT, and the DFT representation of the signal is not compact. In (c), an oversampled DFT of size $K = 5N$ is used; here the sinusoid from (b) can be identified exactly since $\omega_0 = 2\pi(k_0 + 0.4)/N = 2\pi(5k_0+2)/K = 2\pi\kappa_0/K$. In (d), a Hanning window is applied to the signal before the oversampled DFT is carried out. In this figure and in Figure 2.10, filled circles indicate when perfect estimation is achieved; in cases where the estimation is imperfect, the actual signal components are depicted by asterisks.

the signal is not compact. The parameters of the sinusoid can, however, be estimated by interpolation. Using an oversampled DFT is one such approach. a DFT of size $K > N$ is given by

$$X_K[k] \;=\; \alpha_0\, e^{j\omega_0\left(\frac{N-1}{2}\right)}\, e^{-j\pi k\left(\frac{N-1}{K}\right)} \left[\frac{\sin\left(N\left[\frac{\pi k}{K} - \frac{\omega_0}{2}\right]\right)}{\sin\left(\frac{\pi k}{K} - \frac{\omega_0}{2}\right)}\right]. \tag{2.61}$$

In the oversampled case, a sinusoid of frequency

$$\omega_0 \;=\; \frac{2\pi\kappa_0}{K} \tag{2.62}$$

can be identified exactly as a peak in the spectrum as shown in Figure 2.9(c); sinusoids at other frequencies cannot be immediately estimated from the $K$-point DFT. Higher resolution can be achieved, however, by simply choosing a larger $K$.

The spectral representation in Figure 2.9(c) is not compact because using an oversampled DFT corresponds to padding the end of the signal with $K - N$ zeros prior to taking the $K$-point DFT. The signal is then equivalent to a sinusoid of length $K$ time-limited by a window of length $N$, which means that the spectrum corresponds to the $K$-point DFT of a sinusoid of length $K$ circularly convolved with the $K$-point DFT of a rectangular window of length $N$. The time localization provided by the window thus induces a corresponding frequency delocalization.

In STFT filter banks, as mentioned earlier, oversampling in frequency is simply equivalent to adding more filters to the filter bank and decreasing their frequency spacing; this is readily indicated in the following consideration. For an analysis window $w[n]$ of length $N$, the filters in an $N$-channel filter bank are given by

$$h_{k\in\{0,N-1\}}[n] \;=\; w[-n]e^{j2\pi kn/N}. \tag{2.63}$$

In terms of the STFT tiling in Figure 2.1, this corresponds to using a critically sampled DFT for each vertical slice of the tiling. In a $K$-channel filter bank with $K > N$, which corresponds to using an oversampled DFT, the filters are modulated versions of the same prototype window as in the $N$-channel case, namely

$$h_{k\in\{0,K-1\}}[n] \;=\; w[-n]e^{j2\pi kn/K}, \tag{2.64}$$

but the spacing of the channels is now $2\pi/K$, which is less than the $2\pi/N$ spacing in the previous case.

For a single DFT, *i.e.* one short-time spectrum in the STFT, oversampling in frequency corresponds to time-limited interpolation of the spectrum. Other methods of spectral interpolation can also be used to identify the location of the spectral peak; these are generally based on application of a particular window to the original data. Then, the sinusoid can be identified if the shape of the window transform can be detected in the spectrum; the performance of such methods has been considered in the literature for the general case of multiple sinusoids in noise [122, 100, 133]. This matching approach is particularly applicable when a Gaussian window is used since the window transform is then simply a parabola in the log-magnitude spectrum; by fitting a parabola to the spectral data, the location of a peak can be estimated. Such interpolation methods can be coupled with oversampling. An example is given in Figure 2.9(d), in which a Hanning window is applied to the data prior to zero-padding; note that this windowing broadens the main lobe of the spectrum but reduces the sidelobes.

### Two sinusoids

The case of a single sinusoid is of limited interest for modeling musical signals. With a view to understanding the issues involved in modeling complicated signals, the considerations are extended in this section to the case of two sinusoids. It will be indicated by example that the interference of the two components in the frequency domain leads to estimation errors; it is shown to be generally erroneous in multi-component signals to assume that a spectral peak corresponds exactly to a sinusoid in the signal. The reduction of such errors will be used to motivate certain design constraints.

The signal in question will simply be a sum of unit-amplitude, zero-phase sinusoids defined on $n \in [0, N-1]$:

$$x[n] \;=\; e^{j\omega_0 n} \;+\; e^{j\omega_1 n}. \tag{2.65}$$

When $\omega_0$ and $\omega_1$ both correspond to bin frequencies of an $N$-point DFT, both sinusoids can be estimated exactly in the DFT spectrum as indicated in Figure 2.10(a). As shown in Figures 2.10(b) and 2.10(c), the $N$-point DFT cannot identify the sinusoids if either of the frequencies is off-bin; The situation is particularly bleak in Figure 2.10(b), where the two sinusoids are close in frequency.

In the case of a single sinusoid, oversampling was used to improve the frequency resolution. For the case of two closely spaced sinusoids, oversampling does not provide a similar remedy. As depicted in Figure 2.10(c), closely spaced sinusoids in an oversampled DFT appear as a single lobe; neither component can be accurately resolved, and it is inappropriate to identify the spectral peak as a single sinusoid in the signal. Figures 2.10(d) and 2.10(e) show that the resolution of the oversampled DFT tends to improve as the frequency difference increases. Note that in all of the simulations, $\omega_0 = 2\pi\kappa_0/K$ and $\omega_1 = 2\pi\kappa_1/K$ for some integers $\kappa_0$ and $\kappa_1$. This choice of frequencies provides a best-case scenario for the application of oversampled DFTs, and yet various errors still occur; the peaks in the spectrum do not generally correspond to the sinusoids in the signal, so estimation of the sinusoidal components by peak picking is erroneous.

### Resolution of harmonics

As evidenced in Figure 2.10, separation of the spectral lobes improves the ability to estimate the sinusoidal components. This property can be used to establish a criterion for choosing the length of the signal frame $N$ in STFT analysis. A reasonable limiting condition for approximate resolution of two components is that two main lobes appear as separate structures in the spectrum; this occurs when the component frequencies differ by at least half the bandwidth of the main lobe, where the bandwidth is defined here as the distance between the first zero crossings on either side of the lobe. Mathematically, this condition leads to the constraint

$$|\omega_0 - \omega_1| \;\geq\; \frac{2\pi}{N}, \tag{2.66}$$

where the oversampling factor does not appear; oversampling helps in identifying off-bin frequencies that are widely separated, but does not improve the resolution of closely spaced components. In short, the constraint simply states that components must be separated by at least a bin width in an $N$-point DFT to be resolved; this requirement was already suggested in Figure 2.10(b), and will play a further role in the next section. Note that the constraint in Equation (2.66) involves the standard tradeoff between time and frequency

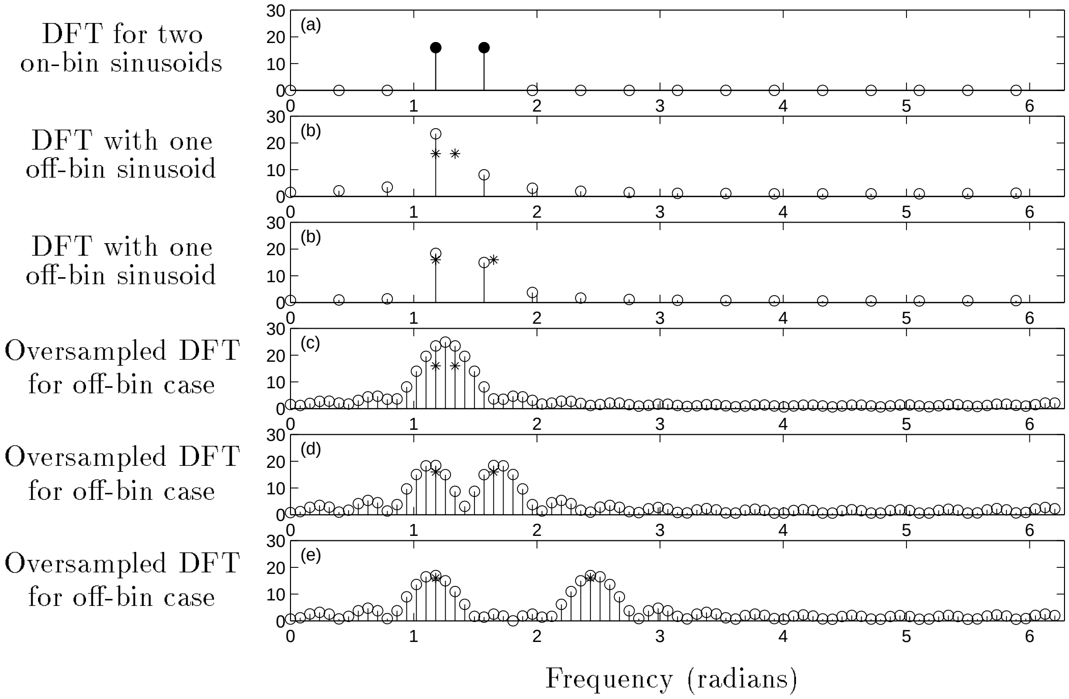

FIGURE 2.10: Estimation of two sinusoids with the DFT. In (a), the sinusoids are at bin frequencies $2\pi k_0/N$ and $2\pi k_1/N$ for $N = 16$, $k_0 = 3$, and $k_0 = 4$; an $N$-point DFT identifies the sinusoids exactly. As in Figure 2.9, filled circles indicate when perfect estimation is achieved; in cases with imperfect estimation, the actual signal components are indicated by asterisks. In (b), $\omega_1$ is moved off-bin to $2\pi(k_0 + 0.4)/N$ as shown by the asterisk; in (c), $\omega_1$ is moved off-bin to $2\pi(k_0 + 1.2)/N$. In either case, the sinusoids are not identified by the DFT. In (d), an oversampled DFT of size $K = 5N$ is used for the sinusoids in (b); these cannot be resolved by oversampling, however. In (e), oversampling is applied for the case in (c); because these sinusoids are separated in frequency, oversampling improves the resolution. The plot in (f) depicts a more extreme case of frequency separation in which the sinusoids can again be reasonably identified. Note that in (d), (e), and (f), the sinusoids cannot be resolved even though their frequencies can be expressed as $2\pi\kappa_0/K$ and $2\pi\kappa_1/K$ for integer $\kappa_0$ and $\kappa_0$; this difficulty results from the interference of the sidelobes in the combined spectrum, or equivalently because the components are not orthogonal as will be explained in Section 2.3.2.

resolution; if $N$ is large, accurate frequency resolution is achieved, but this comes with a time delocalization penalty resulting from using a large window.

The constraint in Equation (2.66) cannot be applied without some knowledge of the expected frequencies in the signal. While this is a questionable requirement for arbitrary signals, it is applicable in the common case of pseudo-periodic signals. The components in the harmonic spectrum of a pseudo-periodic signal are basically multiples of the fundamental frequency, so the constraint can be rewritten as

$$\omega_{\mathrm{fund}} N \;\geq\; 2\pi. \tag{2.67}$$

Note that this constraint can be interpreted in terms of the number of periods of the fundamental frequency,*i.e.* pitch periods of the signal, that occur in the length-$N$ frame; for the components to be resolvable, it is required that at least one period be in the frame. When the $N$-point window spans exactly one period, an $N$-point DFT provides exact resolution of the harmonic components; this observation will play a role in the pitch-synchronous sinusoidal model discussed in Chapter 5.

The formulation of the constraint in Equation (2.67) implicitly assumes the use of a rectangular window. For a Hanning window, the main spectral lobe is twice as wide as that of a rectangular window by construction; as a result, a Hanning window must span two signal periods to achieve resolution of harmonic components. Since Hanning and other similarly constructed windows have been commonly used, it has become a heuristic in STFT analysis to use windows of length two to three times the signal period.

### Modeling arbitrary signals

Analysis based on the DFT has been used in numerous sinusoidal modeling applications [57, 36, 100]. These methods incorporate the constraints discussed above for resolution of harmonics and have been successfully applied to modeling signals with harmonic structure. Furthermore, the approaches have also shown reasonable performance for modeling signals where the sinusoidal components are not resolvable and peak picking in the DFT spectrum provides an inaccurate estimate of the sinusoidal parameters. This issue is examined here.

Consider a signal of the form given Equation (2.65) with component frequencies $\omega_0$ and $\omega_1$ closely spaced as in Figures 2.10(b) and 2.10(d). In this case, peak picking in the oversampled DFT spectrum identifies a peak between $\omega_0$ and $\omega_1$ and interprets this peak as a sinusoid in the signal. At this point, it is assumed that the DFT is oversampled such that $\omega_0 = 2\pi\kappa_0/K$ and $\omega_1 = 2\pi\kappa_1/K$ for integers $\kappa_0$ and $\kappa_1 = \kappa_0 + 2i$, where $i$ is an integer; this condition simply means that there will be an odd number of points in the oversampled DFT between $\kappa_0$ and $\kappa_1$. Then, the peak location found by peak picking is

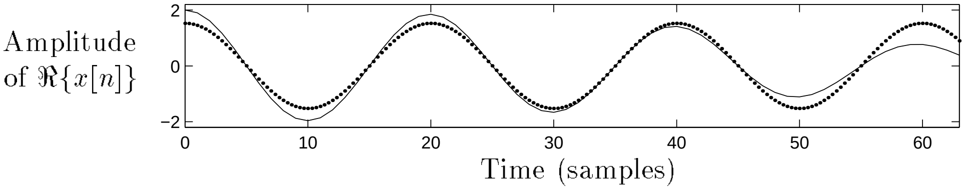

FIGURE 2.11: Modeling a two-component signal via peak picking in the DFT. In the two-component signal of length $N = 64$, the frequencies are at $2\pi\kappa_0/K$ and $2\pi\kappa_1/K$ for $\kappa_0 = 15$, $\kappa_1 = 17$, and $K = 5N$. The sinusoids are closely spaced, so a peak picking process finds only one sinusoid. The signal is indicated by the solid line in the plot; the dotted line indicates the sinusoid estimated by peak picking.

simply given by

$$\omega_p = \frac{\omega_0 + \omega_1}{2} \tag{2.68}$$

$$\Longrightarrow \kappa_p = \frac{\kappa_0 + \kappa_1}{2}. \tag{2.69}$$

When $\kappa_0$ and $\kappa_1$ are related in such a way, the oversampled DFT has a peak midway between $\kappa_0$ and $\kappa_1$ which the analysis interprets as a sinusoid in the signal with frequency $\omega_p$ and with amplitude and phase given respectively by the magnitude of the peak by the phase of the oversampled DFT at the peak frequency. An example of a two-component signal and the signal estimate given by peak picking is indicated in Figure 2.11.

In considering the signal estimate for the case of closely spaced sinusoids, it is useful to rewrite the two-component signal as

$$x[n] = e^{j\left(\frac{\omega_0+\omega_1}{2}\right)n}\left[e^{j\left(\frac{\omega_0-\omega_1}{2}\right)n} + e^{j\left(\frac{-\omega_0+\omega_1}{2}\right)n}\right] = 2\cos\left[(\omega_0 - \omega_1)n/2\right]e^{j\omega_p n}, \tag{2.70}$$

which indicates that the signal can be written as a sinusoid at $\omega_p$ with an amplitude modulation term. In terms of the DFT spectrum, the broad lobe resulting from the overlap of the narrow lobes of the two components can be interpreted as a narrow lobe at a midpoint frequency that has been widened by an amplitude modulation process. It is useful to note the behavior of this modulation for limiting cases: the closer the spacing in frequency, the less variation in the amplitude, which is sensible since the components become identical as $\omega_0 \to \omega_1$; for wider spacing in frequency, the modulation becomes more and more drastic, but this is accompanied by an improved ability to resolve the components. The intuition, then, is that when the components cannot be resolved, the modulation is smooth within the signal frame. This modulation interpretation is not applied in the DFT-based sinusoidal analysis, which estimates the signal components in a frame in terms of constant amplitude sinusoids. As will be discussed in Section 2.4.2, however, the synthesis routine constructs an amplitude envelope for the partials estimated in the frame-to-frame analysis;t this helps to match the amplitude behavior of the reconstruction to that of the signal. In other words, smooth modulation of the amplitude can be tracked by the model.

The example discussed above involves a somewhat ideal case. For one, the formulation is slightly more complicated when the component amplitudes are not equal. Furthermore, when the assumptions previously made about the component frequencies do not hold, the peak picking process becomes more difficult. However, the insights apply to the case of general signals. For arbitrary signals, then, it is reasonable to interpret each lobe in the oversampled DFT as a short-time sinusoid. Given this observation, the partial parameters for a short-time signal frame can be derived by locating major peaks in the DFT magnitude spectrum. For a given peak, the frequency $\omega_q$ of the corresponding partial is estimated as the location of a peak and the phase $\phi_q$ is given by the phase of the spectrum at the peak frequency $\omega_q$. Note that in the frame-rate sinusoidal model, the estimated parameters are designated to correspond to the *center* of the analysis window, so the phase must be advanced from its time reference at the start of the window by adding $\omega_q N/2$. The amplitude $A_q$ of the partial is given by the height of the peak, scaled by $N$ for the case of a rectangular window. This scaling factor amounts to the time-domain sum of the window values, so scaling by $N/2$ is called for in the case of a Hanning window; note that the peak in Figure 2.9(d) is at half the height of the peak in Figure 2.9(d). Further scaling by a factor of $1/2$ is required if the intent is to estimate real sinusoids from a complex spectrum. Also, there is a positive frequency and a negative frequency contribution to the spectrum for this case of real sinusoids, which can result in some spectral interference that may bias the ensuing peak estimation; this is analogous to the estimation errors that occur due to sidelobe interference in the two-component case. While this method is prone to such errors, it is nevertheless useful for signal modeling; the models depicted in later simulations rely on analysis based on oversampled DFTs.

### 2.3.2 Linear Algebraic Interpretation

In the previous section, estimation of the parameters of a sinusoidal model using the DFT was considered. It was shown that this estimation process is erroneous in most cases, but that the errors can be reduced by imposing certain constraints. Here, the estimation problem is phrased in a linear algebraic framework that sheds light on the errors in the DFT approach and suggests an improved analysis.

#### Relationship of analysis and synthesis models

The objective in sinusoidal analysis is to identify the amplitudes, frequencies, and phases of a set of sinusoids that accurately represent a given segment of the signal. This problem can be phrased in terms of finding a compact model using an overcomplete dictionary of sinusoids; the background material for this type of consideration was discussed in Section 1.3. For an $N \times K$ dictionary matrix whose columns are the normalized sinusoids

$$d_k \;=\; \frac{1}{\sqrt{N}} e^{j\omega_k n}, \tag{2.71}$$

where $\omega_k = 2\pi k/K$, the synthesis model for a segment of length $N$ can be expressed in matrix form as

$$x \;=\; D\alpha, \tag{2.72}$$

where $x$ and $\alpha$ are column vectors. Finding a sparse solution to this inverse problem corresponds to deriving parameters for the signal model

$$x[n] \;=\; \frac{1}{\sqrt{N}} \sum_{k=1}^{K} \alpha_k e^{j\omega_k n} \tag{2.73}$$

where many of the coefficients are zero-valued.

In the previous section, analysis for the sinusoidal model using the DFT was considered. The statement of the problem given here, however, indicates that the DFT is by no means intrinsic to the model estimation. In general cases, the exact analysis for an overcomplete model requires computation of a pseudo-inverse of $D$, which is related to projecting the signal onto a dual frame. In deriving compact models, a nonlinear analysis such as a best basis method or matching pursuit is used. Even in the limiting case that $D$ is a basis matrix and the frequencies are known but *not* at frequencies $2\pi k/N$, the DFT is not involved; the model coefficients are given by correlations with the dual basis. The only case in which the DFT is entirely appropriate for analysis of multi-component signals is the orthogonal case where the synthesis components are harmonics at the bin frequencies. It was shown in the previous section, however, that the errors in the DFT analysis are not always drastic. This issue is examined in the next section.

### Orthogonality of components

As stated above, the DFT is only appropriate for analysis when the synthesis components are orthogonal. This explains the perfect analyses shown in Figures 2.9(a) and 2.10(a) for the cases of sinusoids at bin frequencies. The one-component example in Figure 2.9 is not of particular value here, though; even in the general overcomplete case described above, analysis of one-component signals can be carried out perfectly without difficulties. The multi-component case, on the other hand, is problematic and is thus of interest.

Figure 2.10 and the accompanying discussion of frequency separation led to the conclusion that components can be reasonable resolved by peak picking in the DFT spectrum if the components are spaced by at least a bin. Consider two unit-norm sinusoids at different frequencies defined as

$$g_0[n] \;=\; \frac{1}{\sqrt{N}}\, e^{j2\pi\kappa_0 n/K} \quad \text{and} \quad g_1[n] \;=\; \frac{1}{\sqrt{N}}\, e^{j2\pi\kappa_1 n/K}. \tag{2.74}$$

The magnitude of the correlation of these two functions is given by:

$$|\langle g_0, g_1\rangle| \;=\; \frac{1}{N}\left|\frac{\sin\left(\frac{\pi N[\kappa_0-\kappa_1]}{K}\right)}{\sin\left(\frac{\pi[\kappa_0-\kappa_1]}{K}\right)}\right|. \tag{2.75}$$

This function is at a maximum for $\kappa_0 = \kappa_1$, when the sinusoids are equivalent; $|\kappa_0 - \kappa_1| > K/N$, namely separation by more than a bin in an $N$-point spectrum, corresponds to the sidelobe region, where the values are significantly less than the maximum. This insight explains why separation of lobes in the spectrum leads to reasonable analysis results in the DFT approach; when the lobes are separated, the signal components are not highly correlated, *i.e.* are nearly orthogonal. Furthermore, this explains why DFT analysis for the sinusoidal model works reasonably well in cases where the window length is chosen according the constraint in Equation (2.67).

### Frames of complex sinusoids

In discussion of the sinusoidal model, a localized segment of the signal has often been referred to as a frame. Treating the sinusoidal analysis in terms of frames of vectors, then, introduces an unfortunate overlap in terminology. For this discussion, the localized portion of the signal will be assumed to be a segment of length $N$, and the term frame will be reserved to designate an overcomplete family of vectors.

The frame of interest here is the family of vectors

$$d_k \;=\; \frac{1}{\sqrt{N}} e^{j2\pi kn/K} \qquad n \in [0, N-1]. \tag{2.76}$$

If $K = N$, this family is an orthogonal basis and signal expansions can be computed using the DFT. For compact modeling of arbitrary signals, however, the overcomplete case ($K > N$) is more useful. Indeed, the oversampled DFT can be interpreted as a signal expansion based on this family of vectors:

$$X_K[k] \;=\; \sum_{n=0}^{N-1} x[n] e^{-j2\pi kn/K} \tag{2.77}$$

$$\;=\; \sqrt{N}\; \langle d_k, x\rangle. \tag{2.78}$$

The reconstruction can then be expressed as

$$x[n] \;=\; \frac{1}{K}\sum_{k=0}^{K-1} X_K[k] e^{j2\pi kn/K} \tag{2.79}$$

$$= \frac{\sqrt{N}}{K} \sum_{k=0}^{K-1} X_K[k] d_k \tag{2.80}$$

$$= \frac{N}{K} \sum_{k=0}^{K-1} \langle d_k, x \rangle d_k. \tag{2.81}$$

Recalling the earlier discussion of zero-padding, the oversampled DFT can be interpreted as an expansion of a time-limited signal on $[0, N-1]$ in terms of sinusoidal expansion functions supported on the longer interval $[0, K-1]$; this interpretation provides a framework for computing a unique expansion in terms of an orthogonal basis. Equation (2.81), on the other hand, indicates another viewpoint based on the discussion of frames in Section 1.4.2; noting the similarity of Equation (2.81) to Equation (1.26), it is clear that the oversampled DFT corresponds to a signal expansion in a tight frame with redundancy $K/N$.

As discussed in Section 1.4.2, frame expansions of the form given in Equation (1.26) are not generally compact. For the oversampled DFT case, this noncompactness is indicated in the previous section in Figures 2.9 and 2.10. These noncompact expansions do provide perfect reconstruction of the signal, but this is of little use given the amount of data required. Restating the conclusion of the previous section in this framework, it is possible in the DFT case to achieve a reasonable signal approximation using a highly compacted model based on extracting the largest values from the noncompact tight frame expansion. This assertion is verified in Figure 2.11 for a simple example; the shortcoming in this example, however, is that there is an exact compact model in the overcomplete set that the DFT fails to identify. With respect to near-perfect modeling of an arbitrary signal, the shortcoming is that there are compact models that are more accurate than the model derived by DFT peak picking. Arriving at such models, however, is a difficult task as described in Section 1.4.2. It is an open question as to whether the incorporation of such approaches in the sinusoidal model will improve the rate-distortion performance with respect to models based on DFT parameter estimation. Derivation of compact models in overcomplete sets is discussed more fully in Chapter 6, but primarily for the application of constructing models based on Gabor atoms. A method for sinusoidal modeling based on analysis-by-synthesis using an overcomplete set of sinusoids is described in Section 2.3.3.

### Synthesis and modifications

In an overcomplete signal model, the components are necessarily not all orthogonal. As discussed briefly in Section 1.4.4, this results in a difficulty in synthesis of modified expansions. Some additive modifications will correspond to vectors in the null space, and no modification will be manifested in the reconstruction. Further, given that a component can be expressed as a sum of other components, some modifications correspond to cancellation of a desired component, or, in the worst case, cancellation of the entire signal. It is thus important to monitor modifications carried out in overcomplete expansions so as to avoid these pitfalls. A formal consideration of these issues is left as an open issue.

While overcomplete sinusoidal models have been widely used for signal modification, the problems discussed above have not been explicitly discussed in the literature. It will be seen in later sections that the parametric structure of the sinusoidal model allows for resolution of some signal cancellation issues; a specific fix discussed in Section 2.5.2 is that phase matching conditions can be imposed on additive components at similar frequencies to prevent destructive interference. Furthermore, cancellation issues are circumvented to a great extent in applications involving sinusoids separated in frequency; as shown in Equation (2.75), such sinusoids are nearly orthogonal. Synthesis based on nearly orthogonal components of an overcomplete set is well-conditioned with respect to modification, so the sinusoidal model performs well in such scenarios.

### 2.3.3 Other Methods for Sinusoidal Parameter Estimation

A number of alternative methods for estimating the parameters of sinusoidal models have been considered in the literature. A brief review is given below; the focus is placed primarily on methods that introduce substantial model adjustments.

#### Analysis-by-synthesis

In analysis-by-synthesis methods, the analysis is tightly coupled to the synthesis; the analysis is metered and indeed adapted according to how well the reconstructed signal matches the original. Often this is a sequential or iterative process. Consider an example involving spectral peak picking: rather than simultaneously estimating all of the peaks, only the largest peak is detected at first. Then the contribution of a sinusoid at this peak, *i.e.* a spectral lobe, is subtracted from the spectrum, and the next peak is detected; this approach can be used to account for sidelobe interaction. One advantage of this structure over straightforward estimation is that it allows the analysis to adapt to reconstruction errors; these can be accounted for in subsequent iterations. On the other hand, this approach can have difficulties because of its greedy nature.

The matching pursuit algorithm to be discussed in Chapter 6 is an analysis-by-synthesis approach; this notion will be elaborated upon considerably at that point. Here, it suffices to note that analysis-by-synthesis has been applied effectively in sinusoidal modeling, especially in the case where the sinusoidal parameters are estimated directly from the time-domain signal [101]. The particular technique of [101] employs a dictionary of short-time sinusoids and is indeed an example of a method that bridges the gap between parametric and nonparametric approaches. At each stage of the analysis-by-synthesis iteration, the dictionary sinusoid that best resembles the signal is chosen for the decomposition; its contribution to the signal is then subtracted and the process is repeated on the residual. Though it uses a dictionary of expansion functions and should thus be categorized as a nonparametric method according to the heuristic distinctions of Sections 1.3 and 1.4, the algorithm indeed results in a parametric model since the dictionary sinusoids can be readily parameterized.

### Global optimization

The common methods of sinusoidal analysis yield frame-rate signal model parameters. Generally the analysis is independent from frame to frame, meaning that the parameters derived in one frame do not necessarily depend on the parameters of the previous frame; in some cases the estimation is guided according to pitch estimates and models of the signal evolution, but such guidance is generally localized among nearby frames. If the entire signal is considered as a whole in the sinusoidal analysis, a globally optimal set of model parameters can be derived. Such optimization is a highly complex operation which requires intensive off-line computation [107]. This issue is related to the method to be discussed in Section 3.4, in which a slightly restricted global modeling problem is phrased in terms of dynamic programming to reduce the computational cost [134].

### Statistical estimation

A wide variety of methods for estimating the parameters of sinusoidal and quasi-sinusoidal models have been presented in the spectral estimation literature. These differ in the structure of the models; some of these differences include assumptions about harmonicity and the behavior of the partial amplitudes, the effects of underestimating or overestimating the model order, *i.e.* the number of sinusoids in the signal, the presence of noise or other contamination, and the metrics applied to determine the parameters, *e.g.* minimum mean-squared error, maximum likelihood, or a heuristic criterion. Key references for these other methods include [135, 136, 137, 138, 139, 140, 141].

## 2.4 Time-Domain Synthesis

Synthesis for the sinusoidal model is typically carried out in the time domain by accumulating the outputs of a bank of sinusoidal oscillators in direct accordance with the signal model of Equation (2.1). This notion was previously depicted in Figure 2.7; the simple structure of the synthesis bank is given again in Figure 2.12 to emphasize a few key points. First, banks of oscillators have been widely explored in the computer music field as an additive synthesis tool [31, 35, 34]. Early considerations, however, were restricted to synthesis of artificial sounds based on simple parameter control functions since corresponding analyses of natural signals were unavailable and since computational capabilities were limited. The development of analysis algorithms has led to the application of this

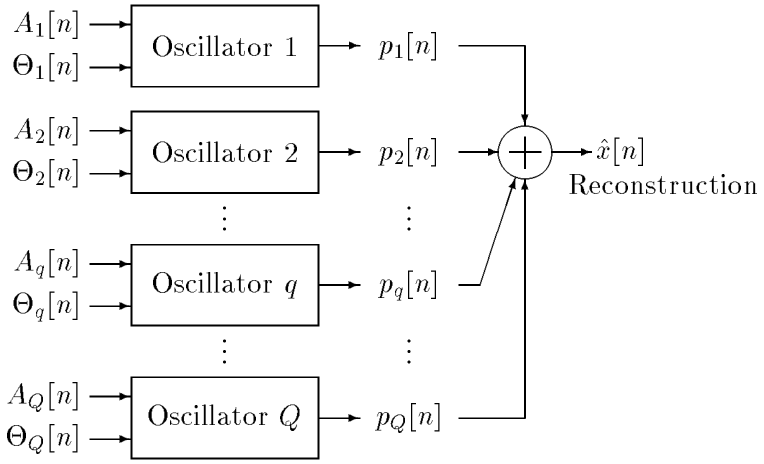

FIGURE 2.12: Time-domain sinusoidal synthesis using a bank of oscillators. The amplitude and phase control functions can be derived using an STFT analysis as depicted in Figure 2.7, or in other ways as described in the text.

approach to modeling and modification of natural signals, and advances in computation technology have enabled such synthesis routines to be carried out in real time [132, 142].

Figure 2.12 also serves to highlight the actual control functions $A_q[n]$ and $\Theta_q[n]$. The output of the $q$-th oscillator is $A_q[n]\cos\theta_q[n]$; this is dictated by sample-rate amplitude and total phase control functions that must be calculated in the synthesis process using the frame-rate (subsampled) analysis data. This process involves two difficulties: *line tracking* and *parameter interpolation*, both of which arise because of the time evolution of the signal and the resultant analysis parameter differences from frame to frame; for instance, the estimated frequencies of the partials change in time as the spectral peaks move. It is of course reasonable that some difficulties should arise, given the intent of generalizing the Fourier series to have arbitrary sinusoidal components; these difficulties are discussed in the following two sections.

### 2.4.1 Line Tracking

The sinusoidal analysis provides a frame-rate representation of the signal in terms of amplitude, frequency, and phase parameters for a set of detected sinusoids in each frame. This analysis provides the sinusoidal parameters, but does not indicate which parameter sets correspond to a given partial. To build a signal model in terms of evolving partials that persist in time, it is necessary to form connections between the parameter sets in adjacent frames. The problem of line tracking is to decide how to connect the parameter sets in adjacent frames to establish continuity for the partials of the signal model. Such continuity is physically reasonable given the generating mechanism of a signal, *e.g.* a vibrating string.

Line tracking can be carried out in a simple successive manner by associating the $q$-th parameter set in frame $i$, namely $\{A_{q,i}, \omega_{q,i}, \phi_{q,i}\}$, to the set in frame $i+1$ with frequency closest to $\omega_{q,i}$ [57]. The tracking starts by making such an association for the pair of parameter sets with the smallest frequency difference across all possible pairs; frequency difference is used as the metric here, but other cost functions, perhaps including amplitude or a predicted rate of frequency change, are of course plausible. Once the first connection is established, the respective parameter sets are taken out of consideration and the process is repeated on the remaining data sets. This iteration is continued until all of the sets in adjacent frames are either coupled or accounted for as *births* or *deaths* – partials that are newly entering or leaving the signal. Generally, there is some threshold set to specify the maximum frequency difference allowed for a partial between frames; rather than coupling a pair of data sets that have a large frequency difference, such instances are treated as a separate birth and death. This tracking is most effective for relatively stationary signal segments; it has difficulty for signal regions where the spectral content is highly dynamic, such as note attacks in music. This breakdown is not so much a shortcoming of the line tracking algorithm as of the signal model itself; a model consisting of smoothly evolving sinusoids is inappropriate for a transient signal.

For complicated signals with many evolving partials, the problem of line tracking is obviously difficult. One important fix, proposed in [36, 100], is the use of backward line tracking when necessary; this technique can be used to track the partials of a note from the sustain region back to their origins in the note attack, which helps with the difficulties previously discussed. Another observation is that line tracking can be aided by considering harmonicity; if the partials are roughly harmonic, the data sets can be coupled more readily than in the general case [57, 36]. A number of more complex methods have been explored in the literature. One noteworthy technique involves using the Viterbi algorithm to find the best set of partial tracks [143, 144]; the cost of a given set of tracks is generally measured by summing the frame-to-frame absolute frequency differences along all of the tracks in the set. This approach finds the set of tracks that has the minimum global cost, *i.e.* the smoothest frequency transitions for the entire set, which is markedly different from the greedy successive track selection algorithm discussed above. This method, which can be cast in the framework of hidden Markov models, has proven useful for sinusoidal modeling of complex sounds [145]. Furthermore, neural networks have been posed as a possible solution to the line tracking problem [146]; nonlinear methods have also proven useful for overcoming some of the difficulties in line tracking [147].

It should be noted that line tracking is sometimes considered part of the analysis rather than synthesis. Then, the model includes a partial index or tag for each parameter set in each frame. The advantage of including this extra data in the representation is that the reconstruction process is simplified such that the synthesis can meet real-time computation constraints. The inclusion is thus useful in cases where the analysis can be performed off-line; for instance, in audio distribution or in real-time signal modification, it is necessary to have a low-complexity synthesis, meaning that high-complexity operations such as line tracking should be lumped with the analysis if possible, even if it does require the inclusion of extra data in the parameterization.

### 2.4.2 Parameter Interpolation

After partial continuity is established by line tracking, it is necessary to interpolate the frame-rate partial parameters $\{A_q, \omega_q, \phi_q\}$ to determine the sample-rate oscillator control functions $A_q[n]$ and $\Theta_q[n]$. Typically, interpolation is done using low-order polynomial models such as linear amplitude and cubic total phase; the specific approach of [57] is presented here, but other interpolation methods have been considered [36, 107, 148, 149, 150]. The partial amplitude interpolation in synthesis frame $i$ is a linear progression from the amplitude in analysis frame $i$ to that in frame $i+1$ and is given by

$$\hat{A}_{q,i}[n] \;=\; A_{q,i} \;+\; (A_{q,i+1} - A_{q,i})\frac{n}{S}, \tag{2.82}$$

where $n = 0, 1, \ldots, S-1$ is the time sample index, and $S$ is the length of the synthesis frame; this frame length is equal to the analysis stride $L$ unless the analysis parameters are intermediately interpolated or otherwise modified to a different time resolution. Note that this amplitude envelope plays a role in modeling sinusoids modulated by slowly varying amplitude envelopes; it was shown in Section 2.3.1 that such partials correspond to components that are not resolved by the DFT analysis. The phase interpolation is given by

$$\hat{\Theta}_{q,i}[n] \;=\; \Theta_{q,i} \;+\; \omega_{q,i}n \;+\; \alpha_{q,i}n^2 \;+\; \beta_{q,i}n^3, \tag{2.83}$$

where $\Theta$ and $\omega$ enforce phase and frequency matching constraints at the frame boundaries, and $\alpha$ and $\beta$ are chosen the make the total phase progression maximally smooth [57]. Such phase and frequency matching constraints are explored in greater detail in Section 2.5.

Interpolation of the phase parameter is clearly more complex than the amplitude interpolation. For efficient synthesis, then, it is of interest to consider more simple models of the phase. Indeed, the experimental observation that the auditory system is relatively insensitive to phase motivates the investigation of models based on amplitude envelopes and low-complexity phase evolution models, thus merging a waveform model with psychoperceptual phenomena in an effort to create a perceptually lossless model. In some cases, this so-called magnitude-only reconstruction can be done transparently; however, transient distortion is increased when the phase is neglected.

In the frequency-domain synthesis algorithm to be discussed in the next section (Section 2.5), the parameter interpolation is not performed directly on the time-domain control functions, but is instead implicitly carried out by an overlap-add process which results in a pseudo-linear amplitude envelope and a transcendental phase interpolation function. These particular interpolation methods will be considered in Section 2.5. The key issue regarding parameter interpolation, however, can be made without reference to a specific interpolation scheme: namely, reconstruction artifacts occur when the behavior of the signal does not match the interpolation model. This idea is revisited in Section 2.6.

## 2.5 Frequency-Domain Synthesis

An alternative to time-domain synthesis using a bank of oscillators is frequency-domain synthesis, in which a representation of the signal is constructed in the frequency domain and the time-domain reconstruction is generated from that representation by an inverse FFT (IFFT) and overlap-add (OLA) process. This approach provides various computational advantages over general time-domain synthesis [102, 132]. Frequency-domain synthesis was described in [57, 150, 151] and more fully presented in [102]. In this section, the algorithm in [102] is explored in detail.

### 2.5.1 The Algorithm

The frequency-domain synthesis algorithm is fundamentally based on the relationship between the DTFT and the DFT and the resulting implications for representing short-time sinusoids. After a brief review of these issues, which are intrinsically connected to the matters discussed in Section 2.3.1, the algorithm is described.

**The DTFT, the DFT, and spectral sampling**

For an N-point discrete-time sequence $x[n]$ defined on the interval $n \in [0, N-1]$, the discrete-time Fourier transform is defined as

$$X\left(e^{j\omega}\right) = \sum_{n=0}^{N-1} x[n]e^{-j\omega n}. \qquad (2.84)$$

Note that the DTFT is inherently $2\pi$-periodic; the signal can be reconstructed from any DTFT segment of length $2\pi$. For the specific interval $[0, 2\pi]$, the equation for signal synthesis is

$$x[n] = \frac{1}{2\pi}\int_0^{2\pi} X\left(e^{j\omega}\right) d\omega, \qquad (2.85)$$

where the interval simply provides the limits for the integral.

The DTFT is a continuous frequency-domain function that represents a discrete-time function; for finite-length signals, there is redundancy in the DTFT representation. The redundancy can be reduced by sampling the DTFT, which is indeed necessary in digital applications. Sampling the DTFT yields a discrete Fourier transform if the samples are taken at uniformly spaced frequencies:

$$X[k] \; = \; X\left(e^{j\omega}\right)\Big|_{\omega=\frac{2\pi k}{K}} \; = \; \sum_{n=0}^{N-1} x[n]e^{-j\frac{2\pi kn}{K}}. \tag{2.86}$$

For $K = N$, the sampled DTFT corresponds to a DFT basis expansion of $x[n]$. If $K < N$, the spectrum is undersampled and time-domain aliasing results as discussed throughout Section 2.2.1. On the other hand, the case $K > N$ corresponds to oversampling of the spectrum; such oversampled DFTs were considered at length in Section 2.3.1 for the application of sinusoidal analysis. If $K \geq N$, the signal can be reconstructed exactly from the DTFT samples using the synthesis formula

$$x[n] \; = \; \frac{1}{K}\sum_{k=0}^{K-1} X[k]e^{j\frac{2\pi kn}{K}}. \tag{2.87}$$

Representations at different spectral sampling rates have a simple relationship if the rates are related by an integer factor; introducing a subscript to denote the size of the DFT,

$$X_K[k] \; = \; X\left(e^{j\omega}\right)\Big|_{\omega=\frac{2\pi k}{K}} \tag{2.88}$$

$$X_M[m] \; = \; X\left(e^{j\omega}\right)\Big|_{\omega=\frac{2\pi m}{M}} \tag{2.89}$$

$$M = \mu K \implies X_M[\mu k] \; = \; X\left(e^{j\omega}\right)\Big|_{\omega=\frac{2\pi\mu k}{\mu K}} \; = \; X_K[k]. \tag{2.90}$$

This relationship will come into play in the frequency-domain synthesis algorithm to be discussed.

The underlying reason that reconstruction can be achieved from the samples of DTFT is that the DTFT is by definition a polynomial function of order $N-1$ (for a signal of length $N$). Then, any $N$ samples specify the DTFT exactly, so the signal can in theory be reconstructed from any $N$ or more arbitrarily spaced samples. The special case of uniform spectral sampling has been of greater interest than nonuniform sampling since it leads to the fast Fourier transform.

### Spectral representation of short-time sinusoids

To carry out frequency-domain synthesis, a spectral representation of the partials must be constructed. This construction is formulated here for the case of a single partial; the extension to multiple partials is developed in the next section.

A short-time sinusoid with amplitude $A_q$, frequency $\omega_q$, and phase $\phi_q$ can be written as

$$p_q[n] \; = \; b[n]A_q e^{j(\omega_q n+\phi_q)}, \tag{2.91}$$

where $b[n]$ is a window function of length $N$. In the frequency domain, this signal corresponds to

$$P_q\left(e^{j\omega}\right) \;=\; B\left(e^{j\omega}\right) * A_q e^{j\phi_q}\delta[\omega-\omega_q] \;=\; A_q e^{j\phi_q} B\left(e^{j(\omega-\omega_q)}\right), \tag{2.92}$$

where $*$ denotes convolution and $B(e^{j\omega})$ is the DTFT of the window $b[n]$. The upshot of this derivation is that the spectrum of a short-time sinusoid windowed by $b[n]$ is the window transform shifted to the frequency of the sinusoid.

For synthesis based on an IDFT of size $K$, the appropriate amplitudes and phases for a $K$-bin spectrum must be determined. This discrete frequency model can be derived via spectral sampling of the DTFT in Equation (2.92):

$$P_q[k] \;=\; A_q e^{j\phi_q}\; B\left(e^{j(\omega-\omega_q)}\right)\Big|_{\omega=\frac{2\pi k}{K}}, \tag{2.93}$$

which corresponds to shifting the window transform $B\left(e^{j\omega}\right)$ to the continuous frequency $\omega_q$ and then sampling it at the discrete bin frequencies $\omega_k = 2\pi k/K$, where $K \geq N$ is required to avoid time-domain aliasing.

Using the above formulation, the short-time sinusoid $p_q[n]$ can be expressed in terms of the $K$-bin IDFT to be used for synthesis:

$$p_q[n] \;=\; \mathrm{IDFT}_K\left\{A_q e^{j\phi_q}\; B\left(e^{j(\omega-\omega_q)}\right)\Big|_{\omega=\frac{2\pi k}{K}}\right\}. \tag{2.94}$$

This representation is depicted in Figure 2.13 for three distinct cases: (1) an unmodulated Hanning window $b[n]$, (2) modulation to a bin frequency of the DFT, and (3) modulation to an off-bin frequency. Note the location of the sample points with respect to the center of the main lobe in each of the cases; in the case of off-bin modulation in (3), the samples are asymmetric about the center. Also note that in (1) and (2) the only nonzero points in the DFT occur in the main lobe since the frequency-domain samples are taken at zero crossings of the DTFT sidelobes. All of the windows in the Blackman-Harris family exhibit this property by construction [122, 123]; it is not a unique feature of the Hanning window. In some applications, this zero-crossing property is useful in that a window can be applied efficiently in the DFT domain by circular convolution [122].

### Spectral motifs

In Equation (2.93) the spectral representation of a short-time sinusoid is computed by evaluating $B\left(e^{j\omega}\right)$ at the frequencies $2\pi k/K - \omega_0$. This computation is prohibitively expensive with regards to real-time synthesis, however, so it is necessary to precompute and tabulate $B\left(e^{j\omega}\right)$ [102, 132]. Such tabulation requires approximating $B\left(e^{j\omega}\right)$ in a discrete form; this approximation, which will be referred to as a spectral *motif* [102], is considered here.

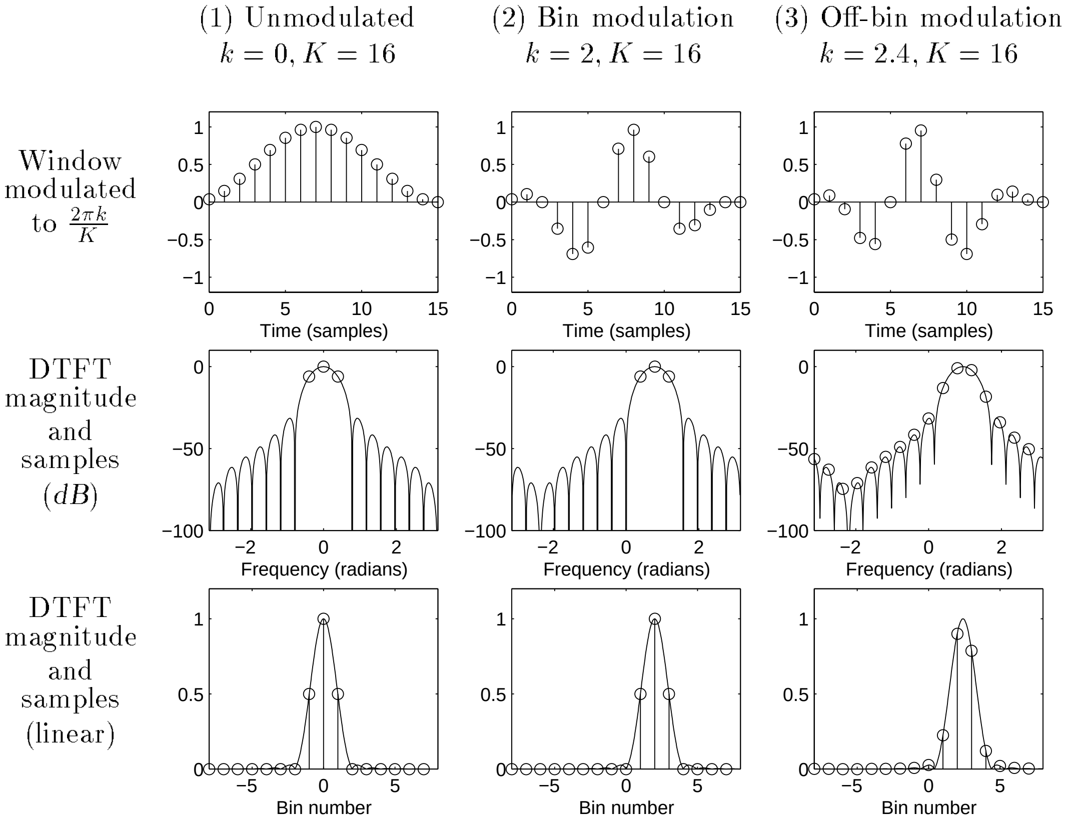

Figure 2.13: A depiction of frequency-domain sampling for spectra of short-time sinusoids. The continuous spectra are the DTFTs of the modulated window functions and the circles indicate the spectral samples corresponding to their DFTs. Case (1) is the unmodulated Hanning window $b[n]$, case (2) involves modulation to the bin frequency $2\pi k/K$ for $k=2$ and $K=16$, and case (3) involves modulation to the off-bin frequency corresponding to $k=2.4$. Note that for $k=0$ and $k=2$, the DFT of the Hanning window consists of only three nonzero points.

A sinusoid at any frequency $\omega_q$ can be represented in the form given in Equation (2.93). Such arbitrary frequency resolution is achieved since $B\left(e^{j\omega}\right)$ is a continuous function and spectral samples can be taken at arbitrary frequencies $2\pi k/K - \omega_q$. In a discrete setting, such resolution can be approximated by representing $B\left(e^{j\omega}\right)$ using a highly oversampled DFT of size $M >> K$; in this framework, the spectral motif is

$$B[m] \;=\; B\left(e^{j\omega}\right)\Big|_{\omega=\frac{2\pi m}{M}}. \tag{2.95}$$

Using such a motif, a sinusoid of frequency $\omega_q = 2\pi m_q/M$ can be represented exactly in a $K$-bin spectrum if $M$ is an integer multiple of $K$, say $M = \mu K$:

$$\begin{aligned} P_q[k] &= A_q e^{j\phi_q}\, B\left(e^{j(\omega-\frac{2\pi m_q}{M})}\right)\Big|_{\omega=\frac{2\pi k}{K}} & (2.96)\\ &= A_q e^{j\phi_q} B\left(e^{j(\frac{2\pi k}{K}-\frac{2\pi m_q}{M})}\right) & (2.97)\\ &= A_q e^{j\phi_q} B\left(e^{j(\frac{2\pi a k}{M}-\frac{2\pi m_q}{M})}\right) & (2.98)\\ &= A_q e^{j\phi_q} B[\mu k - m_q]. & (2.99)\end{aligned}$$

In this way, a spectral representation of a short-time sinusoid is constructed not by directly sampling the DTFT but by sampling the motif, which is itself a sampled version of the DTFT. The frequency resolution is thus limited not by the size of the synthesis IDFT but by the oversampling of the motif; in some other incarnations of frequency-domain synthesis, large IDFTs are required to achieve accurate frequency resolution [57, 150, 151]. In this algorithm, arbitrary frequency resolution can be achieved by increasing the factor $\mu$, provided that enough memory is available for storage of the motif. In music synthesis, however, the resolution limits of the auditory system can be taken into account in choosing the oversampling [102].

Figure 2.14 gives an example of a spectral motif and depicts the resolution issues discussed above. Note that if the frequency of a partial cannot be written as $2\pi m_q/M$, the samples in the shifted motif will not align with the bins of the synthesis IDFT. To account for this, partial frequencies can be rounded; alternatively, linear or higher-order interpolation can be applied to the motif if enough computation time is available. These techniques allow for various tradeoffs between the frequency resolution and the motif storage requirements. Beyond the issue of frequency resolution, a further approximation in the motif-based implementation is also indicated in Figure 2.14. Namely, only the main lobe of $B\left(e^{j\omega}\right)$ is tabulated; the sidelobes are neglected. The result of this approximation is that the spectral representation does not correspond exactly to a sinusoid windowed by $b[n]$; furthermore, each different modulation of the motif actually corresponds to a slightly different window. In practice, these errors are negligible if the window is chosen appropriately [102].

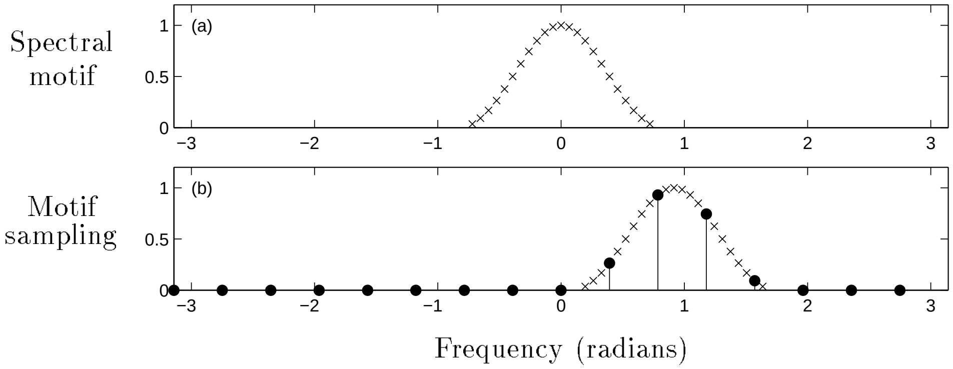

FIGURE 2.14: Spectral motifs in the frequency-domain synthesizer. The motif is the oversampled main lobe of the DTFT of some window $b[n]$, which is precomputed and stored. To represent a partial, the motif is modulated to the partial frequency and then sampled at the bin locations of the synthesis IDFT as shown in (b). If the modulation does not align with the motif samples, the tabulated motif can be interpolated.

In sinusoidal analysis, the issues discussed above lead to the assumption that each lobe in the short-time spectrum of the signal corresponds to a partial. Various caveats involving this assumption were examined in Section 2.3.1; these are not considered further here. The point in this development is simply that a notion that is dual to the sinusoidal analysis applies for frequency-domain synthesis: a partial can be synthesized by inverse transforming an appropriately constructed spectral lobe.

### Accumulation of partials

Since the DTFT and the DFT are linear operations, the spectrum of the sum of partials for the signal model can be constructed by accumulating their individual spectra. Denoting the DTFT for the $i$-th synthesis frame as $\hat{X}\left(e^{j\omega}, i\right)$, and introducing the subscript $i$ to denote the frame to which a partial parameter corresponds, the accumulation of partials for the $i$-th frame is given by

$$\begin{aligned} \hat{X}\left(e^{j\omega}, i\right) &= \sum_{q=1}^{Q} P_{q,i}\left(e^{j\omega}\right) = \sum_{q=1}^{Q} A_{q,i} e^{j\phi_{q,i}} B\left(e^{j(\omega-\omega_{q,i})}\right) \qquad (2.100) \\ &= B\left(e^{j\omega}\right) * \sum_{q=1}^{Q} A_{q,i} e^{j\phi_{q,i}} \delta[\omega - \omega_{q,i}], \qquad (2.101) \end{aligned}$$

which corresponds in the time domain to

$$\hat{x}_i[n] = b[n] \sum_{q=1}^{Q} A_{q,i} e^{j(\omega_{q,i} n + \phi_{q,i})}, \qquad (2.102)$$

which is simply a windowed sum of sinusoids. If $K > N$, the IDFT, implemented as an IFFT for computational efficiency, can be used to generate $\hat{x}_i[n]$ from the sampled spectrum

$$\hat{X}(k,i) \;=\; \hat{X}\left(e^{j\omega},i\right)\Big|_{\omega=\frac{2\pi k}{K}} \;=\; \sum_{q=1}^{Q} A_{q,i}e^{j\phi_{q,i}}\; B\left(e^{j(\omega-\omega_{q,i})}\right)\Big|_{\omega=\frac{2\pi k}{K}}\,. \tag{2.103}$$

This formulation shows that a $K$-bin spectrum for synthesis of a signal segment can be constructed by accumulating sampled versions of a modulated window transform. The result in synthesis is then the sum of sinusoids given in Equation (2.102). To synthesize a sum of real sinusoids, the $K$-bin spectrum can be added to a conjugate-symmetric version of itself prior to the IDFT; note that the window $b[n]$ is assumed real.

As discussed in the previous section, the window transform is represented using a spectral motif. These motifs are modulated according to the partial frequencies from the analysis, and weighted according to the partial amplitudes and phases. The approximations made in the motif representation lead to some errors in the synthesis, though; namely, the motifs for each partial do not exactly correspond to modulated versions of $b[n]$, so the synthesized segment is not exactly a windowed sum of sinusoids. This error can be made negligible, however, by choosing the window appropriately. Noting that the window $b[n]$ is purely a byproduct of the spectral construction, and that it is not necessarily the window used in the sinusoidal analysis, it is evident that the design of $b[n]$ is not governed by reconstruction conditions or the like. Rather, $b[n]$ can be chosen such that its energy is highly concentrated in its main spectral lobe; then, neglecting the sidelobes does not introduce substantial errors. Other considerations regarding the design of $b[n]$ will be indicated in the next section.

### Overlap-add synthesis and parameter interpolation

Given a series of short-time spectra constructed from sinusoidal analysis data as described above, a sinusoidal reconstruction can be carried out by inverse transforming the spectra to create a series of time-domain segments and then connecting these segments with an overlap-add process. This process has distinct ramifications regarding the interpolation of the partial parameters. Whereas in time-domain synthesis the frame-rate data is explicitly interpolated to create sample-rate amplitude and phase tracks, in this approach the interpolation is carried out implicitly by the overlap-add. For reasons to be discussed, it is important to note that the overlap-add can be generalized to include a second window $v[n]$ in addition to $b[n]$; the resultant window will be denoted by $t[n] = b[n]v[n]$. Assuming $t[n]$ is of length $N$ and a stride of $L = N/2$ is used for the OLA, the synthesis of a single partial for one overlap region can be expressed as

$$t[n]A_0e^{j(\omega_0 n+\phi_0)} \;+\; t[n-L]A_1e^{j(\omega_1(n-L)+\phi_1)}, \tag{2.104}$$

where the subscripts 0 and 1 are frame indices, and the subscript $q$ has been dropped for the sake of neatness; the offset of $L$ in the second term serves to adjust its time reference to the start of the window $t[n-L]$. The contributions from the two frames can be coupled into a single magnitude-phase expression; the amplitude evolution of the magnitude-phase form is given by

$$A[n] \;=\; \sqrt{A_0^2 t[n]^2 \;+\; A_1^2 t[n-L]^2 \;+\; 2A_0A_1 t[n]t[n-L]\cos\left[(\omega_0-\omega_1)n+\omega_1 L+\phi_0-\phi_1\right]} \tag{2.105}$$

and the phase is

$$\Theta[n] \;=\; \arctan\left[\frac{A_0 t[n]\sin(\omega_0 n+\phi_0) \;+\; A_1 t[n-L]\sin(\omega_1 n-\omega_1 L+\phi_1)}{A_0 t[n]\cos(\omega_0 n+\phi_0) \;+\; A_1 t[n-L]\cos(\omega_1 n-\omega_1 L+\phi_1)}\right]. \tag{2.106}$$

The region where these functions apply is $n \in [L, N]$, namely the second half of the window $t[n]$ and the first half of $t[n-L]$.

The OLA interpolation functions are clearly more complicated than the low-order polynomials used in time-domain synthesis. The complications arise because the amplitude and frequency evolution are not decoupled as in the time-domain case. The reconstruction in the overlap region is a sum of two sinusoids of different amplitudes and frequencies; these are different since the sinusoidal parameters change from frame to frame for evolving signals. In the OLA interpolation, this parameter difference results in amplitude distortion due to the beating of the different frequencies; furthermore, it results in a transcendental phase function. The parameter interpolation functions in OLA are dealt with further in Section 2.5.2. Here, the discussion will be limited to choosing the synthesis window $t[n]$. This choice will be motivated by adhering to the case of slow signal evolution, where the parameters do not change drastically from one synthesis frame to the next; specifically, the treatment will adhere to the limiting case in which the frequency parameter is assumed constant across frames: $\omega_0 = \omega_1$. This heuristic, coupled with the phase-matching assumptions to be discussed later, leads to a simplification in the amplitude interpolation:

$$A[n] \;=\; A_0 t[n] + A_1 t[n-L]. \tag{2.107}$$

If $t[n]$ is chosen to be a triangular window of length $N$, this overlap-add sum provides linear amplitude interpolation as shown in Figure 2.15. This feature is desirable since it enables the frequency-domain synthesizer to perform similarly to the time-domain method while taking advantage of the computational improvements that result from using the IFFT for synthesis [102, 132].

For the overall OLA window $t[n]$ to be a triangular window, the hybrid window $v[n] = t[n]/b[n]$ must be applied to the IDFT output prior to overlap-add. Thus, the quotient $v[n] = t[n]/b[n]$ must be well-behaved in order for the synthesis to be robust. While $v[n]$ is theoretically a perfect reconstruction window for this OLA process, finite precision

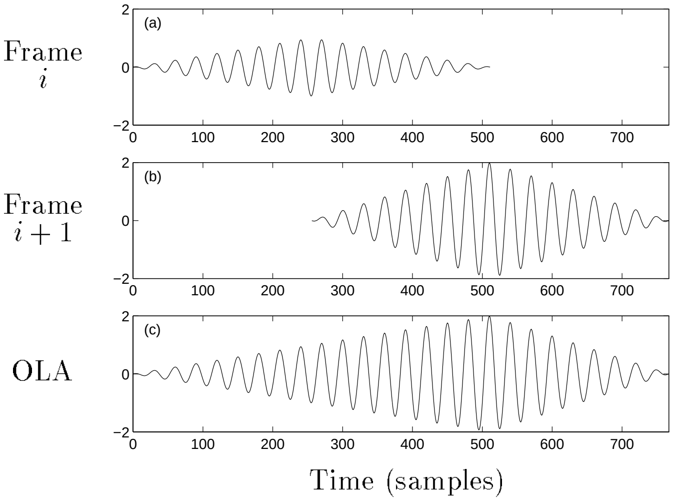

FIGURE 2.15: Overlap-add with a triangular window provides linear amplitude interpolation if the partial frequencies in adjacent frames are equal. Plot (a) shows a triangular-windowed partial of amplitude 1 in synthesis frame $i$, plot (b) shows a partial of amplitude 2 in synthesis frame $i+1$, and plot (c) shows the linear amplitude interpolation resulting from overlap-add of the two frames.

effects may lead to significant errors in the reconstruction if $v[n]$ has discontinuities due to zeros in $b[n]$, for instance. Example of such hybrid windows are given in Figure 2.16 for the case of a Hanning window, a Hamming window, and a Blackman-III window [122]; this shows that a Hanning window is actually unsuitable for this application given the discontinuities at the edges of the hybrid window.

### Frequency-domain synthesis and the STFT

It was shown in Section 2.2.1 that the STFT synthesis can be interpreted as an inverse Fourier transform coupled with overlap-add process. Likewise, the IFFT/OLA process in the frequency-domain synthesizer can be interpreted as an STFT synthesis filter bank. This point of view leads to yet another variation of the block diagrams given in Figures 2.6 and 2.7. In this interpretation, a parametric model is incorporated across all of the bands in the analysis bank as in the sinusoidal model of Figure 2.7; this parametric model includes the sinusoidal analysis and the construction of short-time spectra from the analysis data. Then, the short-time spectra serve as input to a synthesis filter bank, which replaces the oscillator bank used in time-domain sinusoidal synthesis; the filters in the bank are given by $g_k[n] = v[n]e^{j\omega_k n}$ where $v[n]$ is the hybrid window discussed earlier. This structural interpretation of the IFFT/OLA synthesizer is depicted in Figure 2.17; the structure is similar to that used in the STFT modifications discussed in Section 2.2.2.

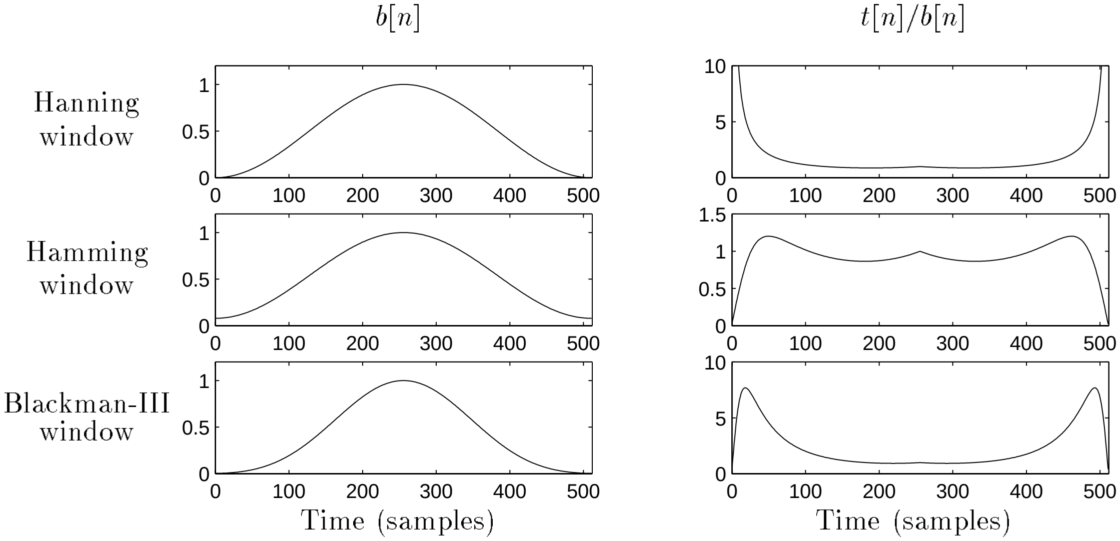

FIGURE 2.16: Overlap-add windows in the frequency-domain synthesizer. The plots in the right column shown $t[n]/b[n]$ when $b[n]$ is a Hanning window, a Hamming window, and a Blackman-III window, respectively.

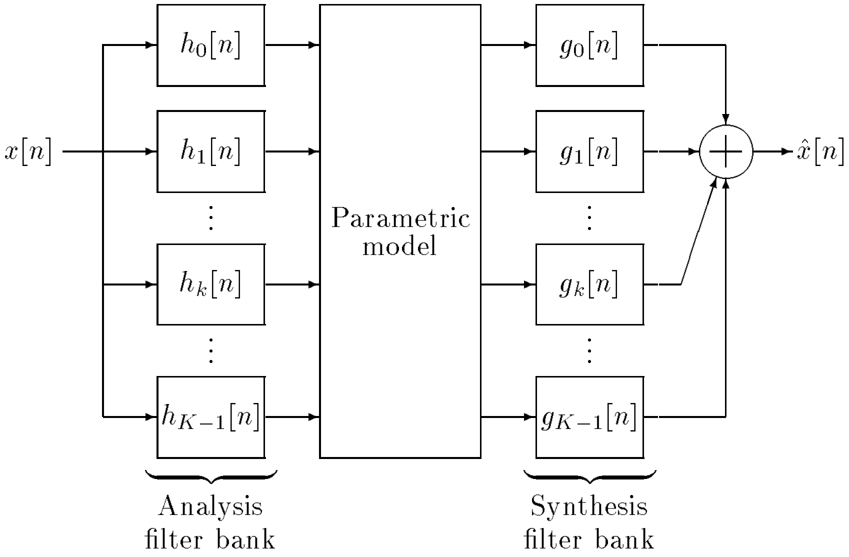

FIGURE 2.17: Block diagram of frequency-domain synthesis for sinusoidal modeling. The parametric model includes the sinusoidal analysis and the construction of short-time spectra from the analysis data. The IFFT/OLA process can be interpreted as an STFT synthesis filter bank.

### 2.5.2 Phase Modeling

In the time-domain synthesizer, low-order polynomial models are used to interpolate the frame-rate parameters to derive sample-rate amplitude and phase functions; this interpolation is carried out explicitly for each partial identified by the line tracking algorithm. In contrast, in the frequency-domain synthesizer the parameter interpolation is carried out implicitly by the overlap-add process; OLA automatically establishes partial continuity without reference to any line tracking method. Line tracking is thus only required for synthesis if a model of continuity is desired for intermediate signal modifications or if the signal is to be reconstructed from the amplitude data only. The latter case is discussed here.

#### Magnitude-only reconstruction and amplitude distortion

Compression can be achieved in the sinusoidal model by discarding the phase data. Such compaction is justifiable in audio applications given the heuristic notion that the ear is insensitive to phase; high-fidelity synthesis can be achieved using only the amplitude and frequency information from the analysis. Such magnitude-only reconstruction, however, relies on imposing sensible phase models that take the frequency evolution into account. In the frequency-domain synthesizer, for instance, ignoring phase relationships in adjacent frames can lead to significant amplitude distortion; consider Equation (2.106) for the simple case $A_0 = A_1 = 1$ with zero phase $\phi_0 = \phi_1 = 0$:

$$A[n] \;=\; \sqrt{t[n]^2 \;+\; t[n-L]^2 \;+\; 2t[n]t[n-L]\cos(\omega_1 L)}. \tag{2.108}$$

The cosine term in this expression can result in highly distorted amplitude envelopes as shown in Figure 2.18. Note that equal amplitudes leads to a worst case scenario since the interfering signals can cancel each other exactly at the midway point in the overlap region.

#### Phase matching

The example in Figure 2.18 shows that neglecting the phase can lead to significant distortion in the OLA synthesis; synthesis with zero phase can result in substantial destructive interference. It is thus necessary to impose a phase model to avoid amplitude distortion artifacts in the reconstruction. One approach to limiting the destructive interference is to match the phases of the interfering sinusoids halfway through the overlap region. This constraint is given by

$$\phi_1 \;=\; \phi_0 + \omega_0\left(\frac{3N}{4}\right) - \omega_1\left(\frac{N}{4}\right), \tag{2.109}$$

where $N = 2L$ is the frame size.

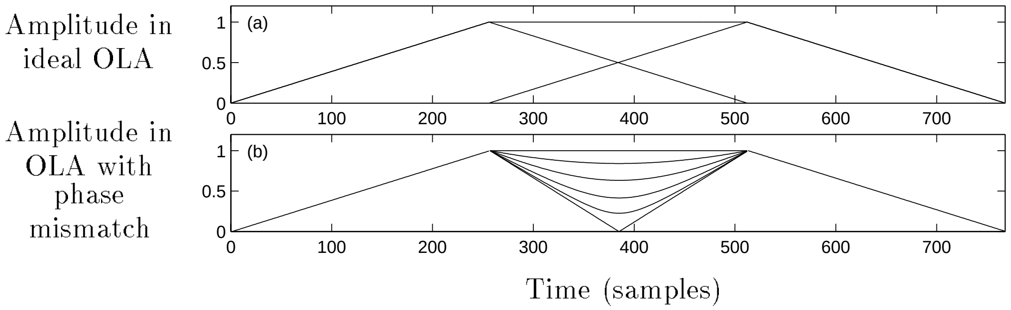

FIGURE 2.18: Plot (a) shows the ideal amplitude envelope for overlap-add with equal amplitudes in adjacent frames; the underlying triangular windows are also shown. Plot (b) shows examples of the amplitude distortion that occurs in the overlap region due to phase mismatch; this example is specifically for the case of frequencies that are equal in adjacent frames as formulated in Equation (2.105), but the effect is general as discussed in the text. In the plot, the phase mismatch $\omega_1 L$ ranges from 0 to $\pi$; for a mismatch of $\pi$, the signals cancel exactly at $n = 3L/4$, halfway through the overlap region.

If the phase matching specified by Equation (2.109) is used, the amplitude envelope, in the equal-amplitude case, becomes a function of the inter-frame frequency difference $\omega_0 - \omega_1$:

$$A[n] \;=\; \sqrt{t[n]^2 \;+\; t[n-L]^2 \;+\; 2t[n]t[n-L]\cos\left[(\omega_0-\omega_1)\left(n-\frac{3N}{4}\right)\right]}. \qquad (2.110)$$

Examples of this amplitude distortion are given in Figure 2.19(a) for $|\omega_0 - \omega_1| = \Delta\pi/N$ with $\Delta \in [0,5]$ and $N = 512$; the corresponding overlap-add phase function $\Theta[n]$ is given for $\Delta \in [0,1,5]$ in Figures 2.19(b,c,d). Note that the amplitude distortion increases as the frequency difference increases and that the phase function is well-behaved, especially for $n = 0$, where it is linear as expected, and for $n = 1$, where the nonlinearity introduced by the frequency change is not pronounced.

To limit the synthesis amplitude distortion characterized in Equation (2.110) and Figure 2.19(a), $N$ can be chosen such that frequency differences in typical signals do not lead to significant distortion. If $N$ is chosen such that

$$\max_{\substack{\text{all frames,}\\ \text{all partials}}} |\omega_{q,i} - \omega_{q,i+1}| \;\leq\; \frac{\pi}{N}, \qquad (2.111)$$

the maximum deviation of the envelope from unity will be less than 2% in the worst case scenario of equal-amplitude partials. Considering this restriction for the case $N = 512$, a $440Hz$ partial at a sampling rate of $44.1kHz$ can double in frequency in about 10 frames, roughly $60ms$, without significant distortion; this rate is suitable for high-quality music synthesis.

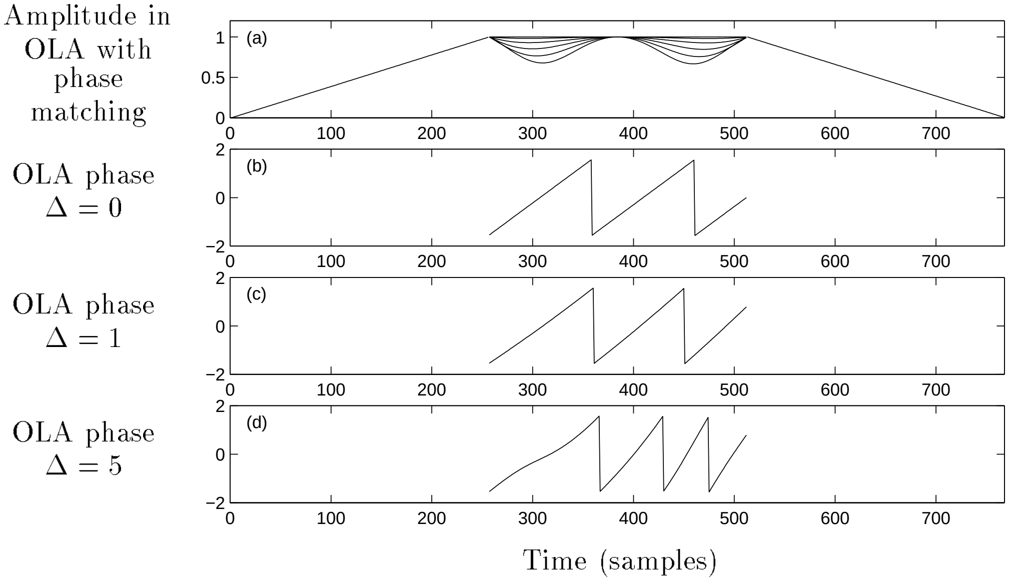

FIGURE 2.19: Parameter interpolation in overlap-add with phase matching. The amplitude distortion in overlap-add is reduced if phase matching is used. If the frequencies in adjacent frames are equal, there is no amplitude distortion and linear interpolation is achieved. In (a), the amplitude distortion is plotted for inter-frame frequency differences for $\Delta\pi/N$, where $\Delta \in [0, 5]$ and $N = 512$. The distortion increases as the frequency difference increases. In plots (b,c,d), the OLA phase function is given for various values of $\Delta$ for $\omega_0 = 5\pi/N$ and $\phi_0 = 0$; the phase is well-behaved.

As stated earlier, the OLA process does not require line tracking if the amplitude and phase data from the analysis are both incorporated in the synthesis. Unlike the time-domain synthesis, which requires tracks for interpolation, the interpolation in OLA is carried out without reference to the signal continuity. However, in cases where compression is achieved by discarding the phase data, it is necessary to use a line tracking algorithm to relate the partials in adjacent frames so that phase matching can be carried out. As shown in this section, in synthesis based on magnitude-only representations it is necessary to incorporate phase modeling to mitigate distortion.

### Frequency matching and chirp synthesis

In addition to phase matching, the synthesis frequencies in adjacent frames can be matched in the overlap region. Such frequency matching can be carried out by synthesizing *chirps* in each frame instead of constant-frequency sinusoids; the chirp rates are determined by a frequency-matching criterion [152, 153]. The caveat in this approach is that the motif must be adjusted to represent a chirp instead of a partial at a fixed frequency, which can be done by precomputing a motif for various chirp rates and interpolating in the precomputed table [152]. Such chirp synthesis, however, has not been shown to be necessary for synthesis of natural signals, so the added cost of tabulation and interpolation is not readily justified. Of course, this conclusion depends on the length of the synthesis windows; if the windows are short enough, the frequency variations from frame to frame will be accordingly small and will not lead to distortion. In a frequency-domain synthesizer with windows on the order of 5 *ms* long, the phase matching described above is sufficient for removing perceptible amplitude distortion in the reconstruction of natural signals.

In Section 2.3.1, the issue of orthogonality of the synthesis components was discussed. Orthogonality was argued to be desirable to avoid destructive interaction in the superposition of components in the signal model; this issue was considered using a geometric framework. Phase modeling can be interpreted in a similar light; considering the windowed partials in adjacent frames as vectors, the phase matching process aligns these vectors in the signal space such that they add constructively instead of destructively.

## 2.6 Reconstruction Artifacts

As discussed in Section 1.5.1, the analysis-synthesis procedure for any signal model has fundamental resolution limits. In the case of the sinusoidal model, the resolution is basically limited by the choice of the frame size and the analysis stride. For long frames, the time resolution is inadequate for capturing signal dynamics such as attack transients; for short frames, on the other hand, the frequency resolution is degraded such that identification of sinusoidal components in the spectrum becomes difficult. The sinusoidal model is thus governed by the same fundamental resolution limits as any time-frequency representation.

In compact models, limitations in time-frequency resolution tend to result in artifacts in the reconstruction. As a result, the analysis-synthesis process yields a nonzero residual. The components of the residual include errors made by the analysis or the synthesis as well as artifacts resulting from basic shortcomings in the model. In the sinusoidal model, for instance, such errors occur if the original signal does not behave in the manner specified by the parameter interpolation used in the synthesis. In addition to the noiselike components discussed in Section 2.1.2, then, the residual in the sinusoidal model contains such model artifacts.

In Section 1.1.2, the perceptual importance of preserving note attacks in music synthesis was discussed. With this in mind, the sinusoidal model artifact that will be focussed on here is pre-echo distortion of signal onsets. This issue was introduced in the example of Figure 2.8; additional examples involving simple synthetic signals are given in Figure 2.20.

The pre-echo depicted in Figures 2.8 and 2.20 is generated by the following mechanism. Before the signal onset, there is an analysis frame in which the signal is not present and no sinusoids are found. For the frame in which the signal onset occurs, various spectral peaks are identified and modeled as sinusoids. The line tracking algorithm interprets these partials as births and forms a track connecting them to zero-amplitude partials in the previous frame, where no spectral peaks were detected. In the reconstruction, then, each of the partials in the onset is synthesized with a linear amplitude envelope as specified by the parameter interpolation model. The result is that the onset is spread into the preceding frame. In general, the birth of a partial in any given frame is delocalized in this manner; in an attack, however, the effect is dramatic because all of the partials are treated in this way simultaneously.

The linear amplitude envelope for a partial onset is clearly visible in the single sinusoid example of Figure 2.20(a,b,c). This example shows not only the delocalization of the attack, but also the introduction of a significant artifact in the residual. Figure 2.20(d,e,f) shows the pre-echo in the sinusoidal model of a harmonic series with three terms; this illustration is given as a precursor to a more complex example involving a natural signal, namely the attack of a saxophone note given in Figure 2.21. The delocalization of the attack degrades the realism of the synthesis, and furthermore introduces an artifact in the residual. These issues will be discussed in detail in the following two chapters; Chapter 3 presents multiresolution extensions of the sinusoidal model intended to improve the localization of transients, and Chapter 4 discusses modeling of the residual. It should be noted here that the frame-rate parameters derived by the sinusoidal analysis can be interpolated to a different rate to achieve data reduction or to match the rate required by the synthesis engine; this process, however, results in additional artifacts due to the

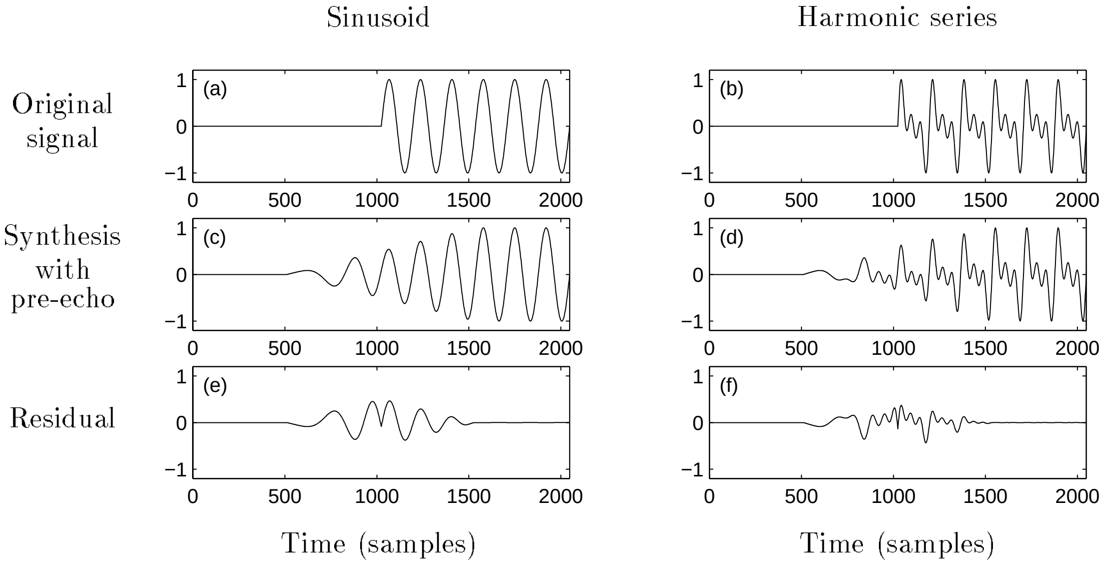

FIGURE 2.20: Pre-echo in the sinusoidal model for two synthetic signals: (a) a simple sinusoid, and (b) a harmonic series. Plots (c) and (d) depict the delocalized reconstructions, and plots (e) and (f) show the respective residuals. Note the pre-echoes and the artifacts near the onset times. frame size of 1024

implicit smoothing of the interpolation.

One approach for preventing reconstruction artifacts is the method described in [101], which accounts for the attack problem by separately modeling the overall amplitude envelope of the signal. The amplitude envelope is imposed on the sinusoidal reconstruction to improve the time localization. This representation, however, is nonuniform in that it relies on independent parametric representations of the envelope and the sinusoidal components. Chapter 3 discusses methods that improve the localization without altering the uniformity of the representation.

## 2.7 Signal Modification

Modifications based on the short-time Fourier transform were discussed in Section 2.2; the difficulty of modifications in such a nonparametric representation was one of the motivations for revamping the STFT into the parametric sinusoidal model. Here, modifications based on the sinusoidal model are dealt with more explicitly. Specifically, time-scaling, pitch-shifting, and cross-synthesis are considered. The treatment here is quite general; formalized details about modifications in a specific version of the sinusoidal model can be found in the literature [93, 101, 154, 94]. Note that the point of this section is not to introduce novel signal modifications, but rather to emphasize that such modifications can be easily realized using the sinusoidal model because of its parametric nature.

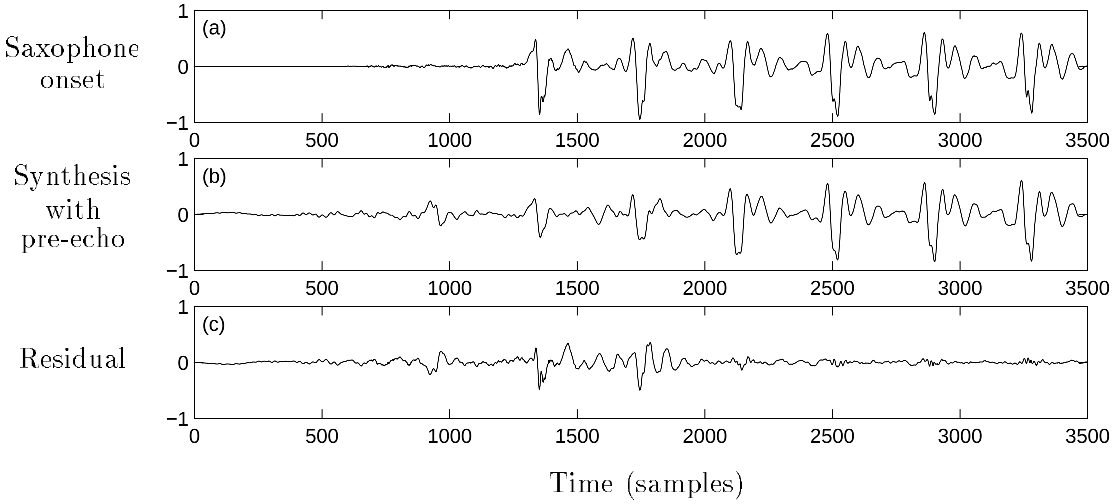

FIGURE 2.21: Pre-echo in the sinusoidal model for a saxophone note: (a) the original, (b) the reconstruction, and (c) the residual.

### 2.7.1 Denoising and Enhancement

The application of denoising deserves mention here inasmuch as the denoising process can be viewed as a signal modification. As discussed, the sinusoidal model is ineffective for representing broadband processes. This shortcoming motivates the inclusion of the stochastic component proposed in [36] to account for musically relevant stochastic features such as breath noise in a flute or bow noise in a violin; these must be incorporated if realistic synthesis is desired. This approach assumes that the original signal is a clean recording of a natural instrument. In cases where the original is a noisy version, the residual in the sinusoidal model basically contains both the noise and the desired stochastic signal features; unless these two noise processes can be somehow separated, this type of residual is not useful for enhancing the signal realism. In these cases, it is generally more desirable to simply not incorporate the residual in the synthesis; in this way, the signal can be denoised via sinusoidal modeling. In addition to denoising, the sinusoidal model has been used for speech enhancement and dynamic range compression. These topics are discussed in the literature [155, 99]

### 2.7.2 Time-Scaling and Pitch-Shifting

In Section 1.5.1, it is proposed that signal modifications can be carried out by modifying the components of a model of the signal. The sinusoidal model is particularly amenable to this approach because the modifications of interest are easy to carry out on sinusoids. For instance, it is simple to increase or decrease the duration of a sinusoid, so if a signal is modeled as a sum of sinusoids, it becomes simple to carry out time-scaling on the entire signal. One caveat to note is that in some time-scaling scenarios it is important to preserve the rate of variation in the amplitude envelope of the signal, *i.e.* the signal dynamics, but this can be readily achieved. This issue is related to the time-scaling of nonstationary signals, in which some signal regions should be time-scaled and some should be left unchanged; for example, for a musical note, which can be most simply modeled as an attack followed by a sustain, time-scale modifications are most perceptually convincing if the time-scaling is carried out only for the sustain region and not for the attack.

Time-scale modifications can also be carried out using approaches traditionally referred to as nonparametric [90]. These involve either STFT magnitude modification followed by phase estimation as discussed earlier, or analyzing the signal for regions, *e.g.* pitch periods, which can be spliced out of the signal for time-scale compression or repeated for time-scale expansion. Computational cost and quality comparisons between such approaches and modifications using the sinusoidal model have not been formally presented, but this is an area of growing interest in the literature and in the electronic music industry [156].

The sinusoidal model allows a much wider range of modifications than standard music synthesizers such as samplers, where the signal is constructed from stored sound segments and modifications are limited by the sample-based representation. For instance, time-scaling in samplers is carried out by upsampling and interpolating the stored signal segments prior to synthesis, but this process is accompanied by a pitch shift. The sinusoidal model can readily achieve time-scaling without pitch-shifting, or the dual modification of pitch-shifting without time-scaling. With regards to pitch modification, a simple form can be carried out by scaling the frequency parameters prior to synthesis, but in voice applications this approach results in unnatural reconstructed speech. Natural pitch transposition can be achieved by interpreting the sinusoidal parameter as a source-filter model and carrying out formant-corrected pitch-shifting, which is discussed below.

### Formant-corrected pitch-shifting

The sinusoidal model parameterization includes a description of the spectral envelope of the signal. This spectral envelope can be interpreted as as a time-varying filter in a source-filter model in which the source is a sum of unweighted sinusoids. In voice applications, the filter corresponds to the vocal tract and the source represents the glottal excitation. This analogy allows the incorporation of an important physical underpinning, namely that a pitch shift in speech is produced primarily by a change in the rate of glottal vibration and not by some change in the vocal tract shape or its resonances. To achieve natural pitch-shifting of speech or the singing voice using the sinusoidal model, then, the spectral envelope must be preserved in the modification stage so as to preserve the formant structure of the vocal tract. The pitch-scaling is carried out by scaling the frequency parameters of the excitation sinusoids and then deriving new amplitudes for these pitch-scaled sinusoids by interpolating from the spectral envelope. This approach allows for realistic pitch transposition.

#### Spectral manipulations

In addition to formant-corrected pitch-shifting, the source-filter interpretation of the sinusoidal model is useful for a variety of spectral manipulations. In general, any sort of time-varying filtering can be carried out by appropriately modifying the spectral envelopes in the parametric sinusoidal model domain. For instance, the formants in the spectral envelope can be adjusted to yield gender modifications; by moving the formants down in frequency, a female voice can be transformed into a male voice, and vice versa [157]. Also, the amplitude ratios of odd and even harmonics in a pitched signal can be adjusted. These modifications are related to methods known as cross-synthesis, which is considered further in the following section.

### 2.7.3 Cross-Synthesis and Timbre Space

Time-scaling and pitch-shifting modifications are operations carried out a single original signal; the term *cross-synthesis* refers to methods in which a new signal is created via the interactions of two or more original signals. A common example of cross-synthesis is based on source-filter models of two signals; as exemplified in the previous section, useful mixture signals can be derived by using the source from one model and the filter from the other, for instance exciting the vocal tract filter estimated from a male voice by the glottal source estimated from a female voice. Such cross-synthesis has been experimented with in music recording and performance; one of the early examples of cross-synthesis in popular music, mentioned in Chapter 1, is the cross-synthesized guitar in [54], in which the signal from an electric guitar pickup is used as an excitation for a vocal tract filter, resulting in a guitar sound with a speech-like formant structure, the percept of which is a “talking” guitar.

Parametric representations enable a wide class of cross-synthesis modifications. This notion is especially true in the sinusoidal model since the parameters directly indicate musically important signal qualities such as the pitch as well as the shape and evolution of the spectral envelope. One immediate example of a modification is interpolation between the sinusoidal parameters of two sounds; this yields a hybrid signal perceived as a coherent merger of the two original sounds, and not simply a cross-fade or averaging. This type of modification has recently received considerable attention for the application of image *morphing*, which is carried out by parameterizing the salient features of an original image and a target image (such as edges or prominent regions) and creating a map between these parametric features that can be traversed to synthesize a morphed image [158].

Such morphing has also been used in the audio domain to carry out modifications based on the parametric representation provided by the *spectrogram*, *i.e.* the squared magnitude of the STFT [129].

In the fields of psychoacoustics and computer music, it has been of interest to categorize instrumental sounds according to their location in a perceptual space. For instance, the clarinet and the bassoon would be fairly close together in this space, while the piano or guitar would not be nearby. Such categorization is referred to as *multidimensional scaling* [35, 34, 108]. It has been observed that *timbre*, which corresponds loosely to the evolution and shape of the spectral envelope, is an important feature in subjective evaluations of the similarity of sounds; if two sounds have the same timbre, they are generally judged to be similar [108]. Because the parameters of the sinusoidal model capture the behavior of the spectral envelope, *i.e.* the timbre of the sound, the sinusoidal representations of various sounds can be used to situate the sounds in a *timbre space*, which can then be explored in musically meaningful ways by interpolating between the parameter sets. This interpretation of a parametric timbre space as a musical control structure has been the focus of recent work in computer music [108].

## 2.8 Conclusion

In this chapter, the nonparametric short-time Fourier transform was discussed extensively. It was shown that the STFT can be interpreted as a modulated filter bank in which the subband signals can be likened to the partials in a sinusoidal signal model. It was further shown that more compact models can be achieved by parameterizing these subband signals to account for signal evolution. This idea is fundamental to the sinusoidal model, which can be viewed as a parametric extension of the STFT; incorporating such parameterization leads to signal adaptivity and compact models. Various analysis issues for the sinusoidal model were considered, and both time-domain and frequency-domain synthesis methods were discussed. Since the sinusoidal model is parametric, any of these analysis-synthesis methods inherently introduce some reconstruction artifacts, but these come with the benefits of compaction and modification capabilities. Minimization of such artifacts by multiresolution methods is discussed in Chapter 3, and modeling of the residual is examined in Chapter 4.

# Chapter 3

# Multiresolution Sinusoidal Modeling

As indicated in the previous chapter, the standard sinusoidal model has difficulty modeling broadband processes – both noiselike components and time-localized transient events such as attacks. Thus, such broadband processes appear in the residual of the sinusoidal analysis-synthesis. A perceptual model for noiselike components will be presented in Chapter 4; that representation, however, is inadequate for time-localized events such as attack artifacts, so it is necessary to consider ways to prevent these events from appearing in the residual. In this chapter, the sinusoidal model is reinterpreted in terms of expansion functions; the structure of these expansion functions both indicates why the model breaks down for time-localized events and suggests methods to improve the model by casting it in a multiresolution framework. Two approaches are considered: applying the sinusoidal model to filter bank subbands, and using signal-adaptive analysis and synthesis frame sizes. These specific methods are discussed after a consideration of multiresolution as exemplified by the discrete wavelet transform.

## 3.1 Atomic Interpretation of the Sinusoidal Model

The partials in the sinusoidal model can be interpreted as expansion functions that comprise an additive decomposition of the signal; this perspective provides a conceptual framework for several considerations of sinusoidal modeling that have been presented in the literature [97, 159, 160]. With this notion as a starting point, the sinusoidal model is here interpreted as a time-frequency atomic decomposition. This interpretation sheds some light on the fundamental modeling issues, and indicates a connection between sinusoidal modeling and granular analysis-synthesis.

As discussed in the previous chapter, in the time-domain synthesis approach the partials are generated at the synthesis stage by interpolating the frame-rate analysis parameters using low-order polynomials. Figure 3.1 depicts a typical partial $A_q[t]\cos\Theta_q[t]$ synthesized using linear amplitude interpolation and cubic total phase as formulated in

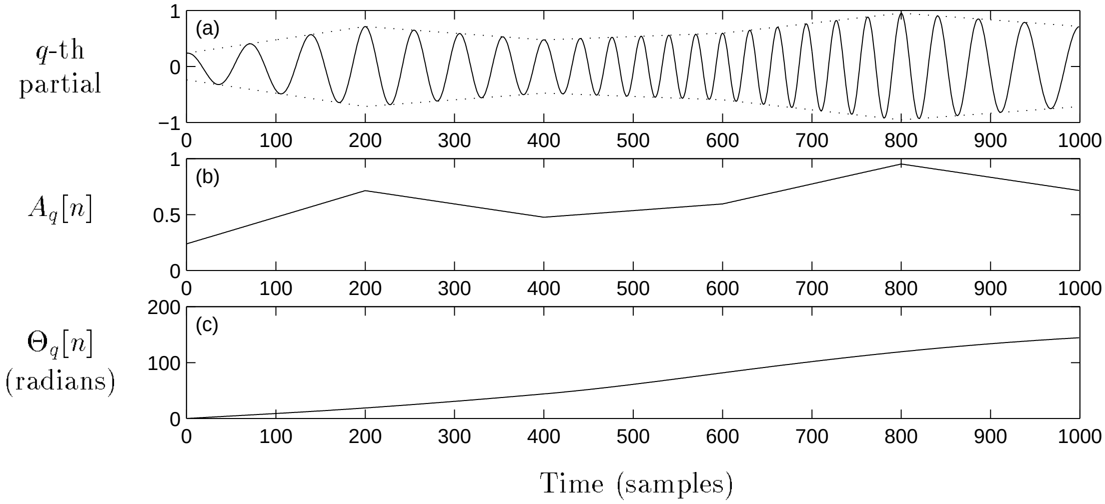

FIGURE 3.1: A typical partial in the sinusoidal model (a) with a linear amplitude envelope (b) and cubic total phase (c).

[57]. In the next section, this example is used to indicate the aforementioned granular interpretation of the sinusoidal model.

The atomic interpretation of the sinusoidal model stems from considering the frame-to-frame nature of the approach. The model given in Equation (2.1), namely

$$x[n] \;\approx\; \hat{x}[n] \;=\; \sum_{q=1}^{Q[n]} p_q[n] \;=\; \sum_{q=1}^{Q[n]} A_q[n]\cos\Theta_q[n], \tag{3.1}$$

can be recast into an expression that incorporates the synthesis frames, which are indexed by the subscript $j$:

$$x[n] \;\approx\; \hat{x}[n] \;=\; \sum_{q=1}^{Q[n]} p_q[n] \;=\; \sum_j \sum_{q=1}^{Q[n]} p_{q,j}[n] \;=\; \sum_j \sum_{q=1}^{Q[n]} A_{q,j}[n]\cos\Theta_{q,j}[n], \tag{3.2}$$

where $p_{q,j}[n]$ denotes the time-limited portion of the $q$-th partial that corresponds to the $j$-th synthesis frame. The time-domain sinusoidal synthesis can thereby be viewed as a concatenation of non-overlapping synthesis frames, each of which is a sum of localized partials. Each of the components $p_{q,j}[n]$ in Equation (3.2) is time-localized to a synthesis frame and frequency-localized according to the function $\Theta_{q,j}[n]$. Thus, a sinusoidal model of a signal can be interpreted as an atomic decomposition given by

$$x[n] \;\approx\; \sum_{q,j} p_{q,j}[n], \quad \text{where} \quad p_{q,j}[n] \;=\; A_{q,j}[n]\cos\Theta_{q,j}[n] \tag{3.3}$$

as indicated in Equation (3.2). For the specific interpolation models discussed in Section 2.4.2, the sinusoidal model derives a signal expansion in terms of atoms with linear amplitude and cubic phase. An example of this atomic decomposition is depicted in Figure

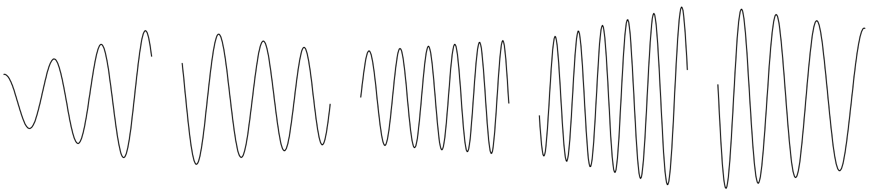

FIGURE 3.2: The partial depicted in Figure 3.1 can be decomposed into these linear amplitude, cubic phase time-frequency atoms. This decomposition suggests an interpretation of the sinusoidal model as a method of granular analysis-synthesis in which the grains are connected in an evolutionary fashion.

3.2; the atoms correspond to the partial of Figure 3.1. Note that the atoms are generated using parameters extracted from the signal and are thus signal-adaptive. In this sense, the sinusoidal model can be interpreted as a method of granular analysis-synthesis; by its parametric nature, it overcomes the limitations of the STFT or phase vocoder with respect to granulation.

In this atomic interpretation of the sinusoidal model, it should be noted that the atoms are connected from frame to frame in accordance with a notion of signal continuity or *evolution*. This connectivity results in partials that persist meaningfully in time. The atoms are not disparate events in time-frequency but rather interlocking pieces of a cohesive whole.

### 3.1.1 Multiresolution Approaches

The atomic interpretation of the sinusoidal model indicates why the model has difficulties representing transient events such as note attacks. Each atom in the decomposition spans an entire synthesis frame; the time *support* or span is the same for every atom. The result of this fixed resolution is that events that occur on short time scales are not well-modeled; this problem is analogous to the difficulty that a Fourier transform has in modeling impulsive signals. In addition to the limitations that result from the fixed time support of the atoms, however, the sinusoidal model also has time-localization limitations because of the frame-to-frame interpolation of the partial parameters as discussed in Section 2.6. The sinusoidal model thus delocalizes transient events in two ways: a transient is spread across a synthesis frame because of the fixed time resolution of the model expansion functions; furthermore, a transient bleeds into neighboring frames due to the interpolation process.

The time-localization shortcomings of the sinusoidal model can be remedied by applying a multiresolution framework to the model. Fundamentally, such approaches are motivated by the atomic interpretation of the model: atoms with constant time support are inadequate for representing rapidly varying signals, so it is necessary to admit atoms with a variety of supports into the decomposition. To this point it has been implied that shorter atoms are of interest, but it should be noted that in some cases it is also useful to lengthen the time support of the atoms. In regions where a signal is well-modeled by a sum of sinusoids, lengthening the frames improves the frequency resolution of the analysis and can thus improve the model; furthermore, long frames are useful for coding efficiency. Incorporating a diverse set of time supports allows for flexible tradeoffs between time and frequency resolution.

As in other sections of this thesis, the focus in this chapter will be on time resolution and pre-echo distortion. Pre-echo results from both of the localization limitations: within a frame and across frames. The first issue is addressed by using short atoms directly at an attack, and the latter by incorporating shorter atoms in the neighborhood to limit the spreading.

There are two distinct approaches by which expansion functions with a variety of time supports can be admitted into the decomposition. In methods based on filter banks, subband filtering is followed by sinusoidal modeling of the channel signals with long frames for low-frequency bands and short frames for high-frequency bands. In time-segmentation methods, the frame size is varied dynamically based on the signal characteristics; short frames are used near transients and long frames are used for regions with stationary behavior. These methods are discussed in Sections 3.3 and 3.4, respectively.

The multiresolution sinusoidal models to be considered incorporate the time-frequency localization advantages of wavelet-based approaches while preserving the flexibility provided by the parametric nature of the sinusoidal model. Since multiresolution and wavelets are intrinsically related, these topics are examined in the next section as a prerequisite to further discussion of multiresolution methods in sinusoidal modeling.

## 3.2 Multiresolution Signal Decompositions

The basic concept of multiresolution was discussed in Section 1.5.1. Here, the issue is developed further; this development is based on the discrete wavelet transform, which is inherently connected to the notion of multiresolution [2, 79].

### 3.2.1 Wavelets and Filter Banks

Wavelets and multiresolution are intrinsically related. For this chapter, wavelets serve as a framework for considering multiresolution as well as the relationship between atomic and filter bank models; an understanding of the wavelet transform will also be useful for future considerations, particularly those of Chapter 5. The focus here will be on the discrete wavelet transform (DWT) and not the related continuous wavelet transform

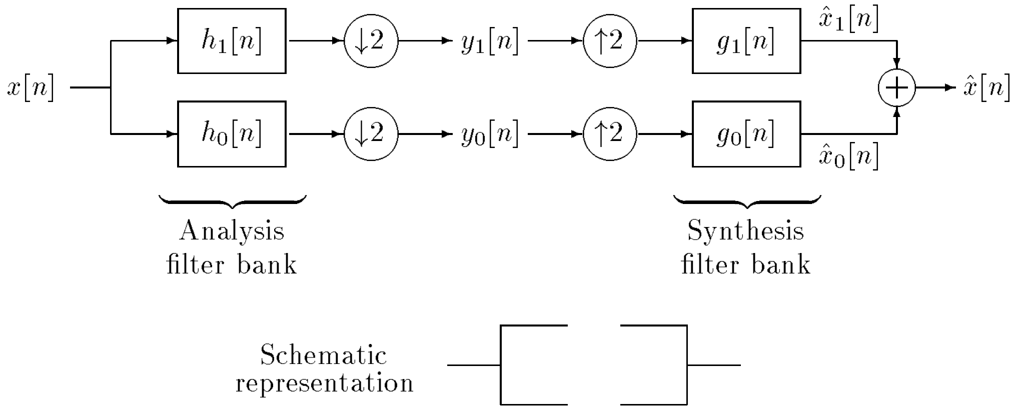

FIGURE 3.3: Critically sampled perfect reconstruction two-channel filter banks having this structure can be used to derive the discrete wavelet transform. In the literature, such a structure is often depicted with the simple line drawing shown. In many applications of such structures, $h_0[n]$ and $h_1[n]$ are respectively a lowpass and a highpass filter; likewise for $g_0[n]$ and $g_1[n]$.

(CWT); for a treatment of the CWT, the reader is referred to [2]. This treatment is not intended as an exhaustive review of wavelet theory but rather as a discussion of wavelets with a view to understanding multiresolution and related signal modeling issues. The treatment is restricted primarily to conceptual matters here; various mathematical details are provided in Appendix A.

### Two-channel critically sampled perfect reconstruction filter banks

The discrete wavelet transform can be derived in terms of critically sampled two-channel perfect reconstruction filter banks such as the one shown in Figure 3.3. The condition for perfect reconstruction can be readily derived in terms of the $z$-transforms of the signals and filters; details of the derivation are given in Appendix A. The resulting constraints on the filters can be summarized as:

$$G_i(z)H_j(z) \ + \ G_i(-z)H_j(-z) \ = \ 2\delta[i-j]. \tag{3.4}$$

In the next section, this condition leads to an interpretation of the filter bank in terms of a biorthogonal basis.

### Perfect reconstruction and biorthogonality

By manipulating the perfect reconstruction condition in (3.4), it can be shown that a perfect reconstruction filter bank derives a signal expansion in a biorthogonal basis; the basis is related to the impulse responses of the filter bank. This relationship is of particular interest in that it establishes a connection between the filter bank model and the atomic model that underlie the discrete wavelet transform.

A full mathematical treatment of this issue is given in Appendix A; the result is simply that the perfect reconstruction condition given in Equation (3.4) can be expressed in the time domain as

$$\langle g_i[k], h_j[2n-k]\rangle \;=\; \delta[n]\delta[i-j], \tag{3.5}$$

or equivalently as

$$\langle h_i[k], g_j[2n-k]\rangle \;=\; \delta[n]\delta[i-j]. \tag{3.6}$$

The above expressions show that the impulse responses of the filters, with one of the impulse responses time-reversed as indicated, constitute a pair of biorthogonal bases for discrete-time signals (with finite energy), namely the space $l^2(z)$; the time shift of $2n$ in the time-reversed impulse response arises because of the subsampling of the channel signals. Note that real filters have been implicitly assumed; for complex filters, the first terms in the inner product expressions would be conjugated. Also note that the analysis and synthesis filter banks are mathematically interchangeable; this symmetry is analogous to the equivalence of left and right matrix inverses discussed in Section 1.4.1.

The result given above indicates that perfect reconstruction and biorthogonality are equivalent conditions. In the next section, this insight is used to relate filter banks and signal expansions.

### Interpretation as a signal expansion in a biorthogonal basis

Since the impulse responses of a perfect reconstruction filter bank are related to an underlying biorthogonal basis, it is reasonable to consider the time-domain signal expansion carried out by the two-channel filter bank. Using the notation given in Figure 3.3, the output of the filter bank can be expressed as follows; more details of the derivation are given in Appendix A:

$$\begin{aligned}
\hat{x}[n] &= \hat{x}_0[n] \;+\; \hat{x}_1[n] && (3.7)\\
&= \sum_k y_0[k]g_0[n-2k] \;+\; \sum_k y_1[k]g_1[n-2k] && (3.8)\\
&= \sum_k \langle x[m], h_0[2k-m]\rangle\, g_0[n-2k] \;+\; \sum_k \langle x[m], h_1[2k-m]\rangle\, g_1[n-2k] && (3.9)\\
&= \sum_{i=1}^{2}\sum_k \langle x[m], h_i[2k-m]\rangle g_i[n-2k]. && (3.10)
\end{aligned}$$

Introducing the notation

$$g_{i,k}[n] \;=\; g_i[n-2k] \quad \text{and} \quad \alpha_{i,k} \;=\; \langle x[m], h_i[2k-m]\rangle, \tag{3.11}$$

the signal reconstruction can be clearly expressed as an atomic model:

$$\hat{x}[n] \;=\; \sum_{i\in\{1,2\},k} \alpha_{i,k} g_{i,k}[n]. \tag{3.12}$$

The coefficients in the atomic decomposition are derived by the analysis filter bank, and the expansion functions are time-shifts of the impulse responses of the synthesis filter bank. As noted earlier, the filter banks are interchangeable; the signal could also be written as an atomic decomposition based on the impulse responses $h_i[n]$. In any case, the atoms in the signal model correspond to the synthesis filter bank.

It has thus been shown that filter banks compute signal expansions. Indeed, any critically sampled perfect reconstruction filter bank implements a signal expansion in a biorthogonal basis, and any filter bank that implements a biorthogonal expansion provides perfect reconstruction; biorthogonality and perfect reconstruction are equivalent conditions [2]. At this point, however, the notion of multiresolution has not yet entered the considerations; the atoms in the decomposition of Equation (3.12) do not have multiresolution properties. In the next section, it is shown that multiresolution can be introduced by iterating two-channel filter banks. Such iteration is fundamental to implementations of wavelet packets and the discrete wavelet transform.

### Tree-structured filter banks and wavelet packets

A wide class of signal transforms, known as wavelet packets, are based on the observation that a perfect reconstruction filter bank with a tree structure can be derived by iterating two-channel filter banks in the subbands. Examples of such tree-structured filter banks are depicted in Figure 3.4. For this treatment, it is important to note that the filters $H_0(z)$ and $H_1(z)$ are generally a lowpass and a highpass, respectively, and likewise for $G_0(z)$ and $G_1(z)$; this lowpass-highpass filtering in the constituent two-channel filter banks leads to spectral decompositions such as those depicted in Figure 3.4 for the given tree-structured filter banks. Frequency-domain interpretations of aliasing cancellation and signal reconstruction based on this lowpass-highpass structure are given in [2, 20].

Arbitrary tree-structured filter banks that achieve perfect reconstruction can be constructed by iterating two-channel perfect reconstruction filter banks; indeed, the filter trees can be made to adapt to model nonstationary input signals while still satisfying the reconstruction constraint [60]. In this treatment, the primary issue of interest is the manner in which iteration of two-channel subsampled filter banks leads to multiresolution. The basic principle is that a two-channel filter bank splits its input spectrum into two bands and the ensuing downsampling spreads each band such that the subband signals are again full band (considered at the subsampled rate); this successive halving leads to the spectral decompositions given in Figure 3.4 for the specific filter banks shown. The spectral decompositions indicate multiresolution in frequency, which is inherently

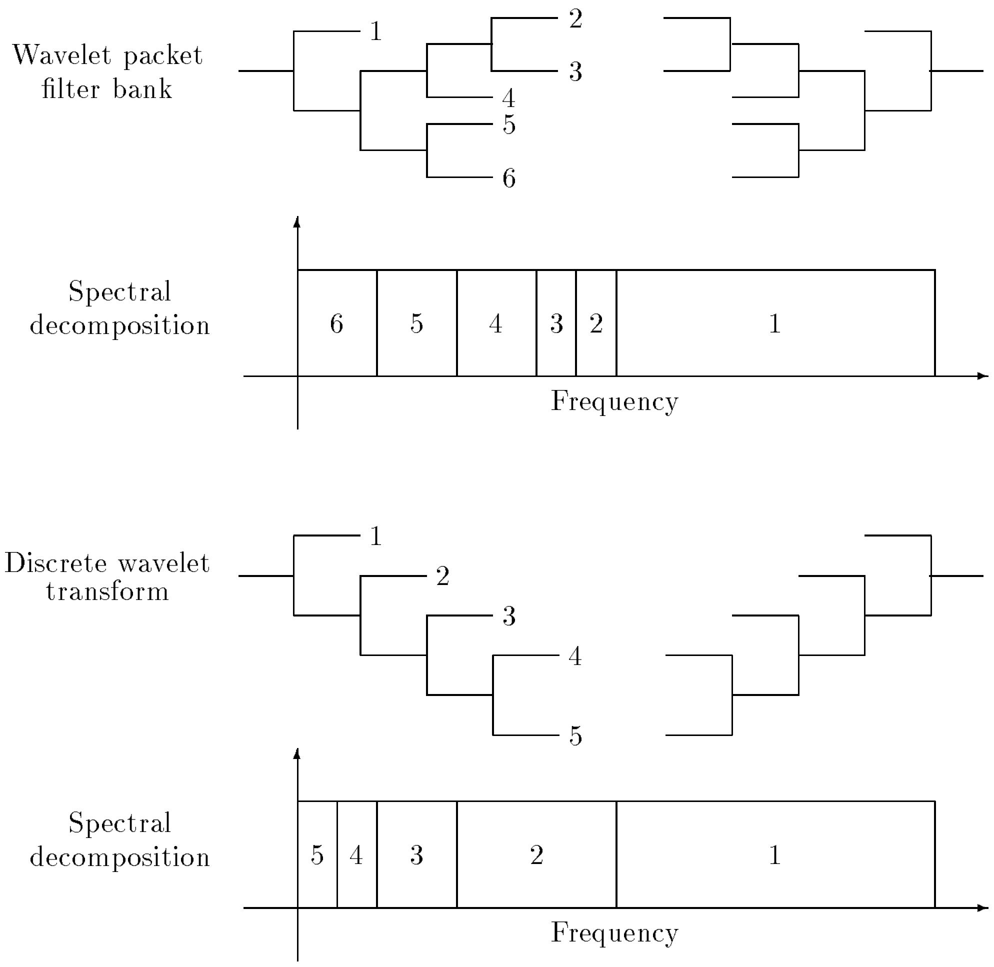

FIGURE 3.4: Tree-structured filter banks that satisfy the perfect reconstruction condition can be constructed by iterating two-channel perfect reconstruction filter banks. Such iteration is fundamental to the discrete wavelet transform as well as arbitrary wavelet packet filter banks. These iterated filter banks provide multiresolution analysis-synthesis as suggested by the indicated spectral decompositions. Note that the discrete wavelet transform derives an octave-band decomposition of the signal.

coupled to multiresolution in time by the principle that to increase frequency resolution, it is necessary to decrease time resolution. The connection is immediate: the narrowest spectral bands correspond to the deepest levels of iteration; each iteration involves a convolution, which spreads out the time resolution of the overall branch, so the subbands that are most localized in frequency are least localized in time.

The brief description of multiresolution in tree-structured filter banks suggests why such methods might prove useful for processing arbitrary signals, especially if the filter bank is made adaptive; application examples include compression [41, 60] and spectral estimation [161]. Rather than focusing on such arbitrary tree-structured filter banks here, however, additional developments of the multiresolution concept will be formulated for the specific case of the discrete wavelet transform. As noted in Figure 3.4, the discrete wavelet transform corresponds to successive iterations on the lowpass branch.

### The discrete wavelet transform

The discrete wavelet transform is perhaps the most common example of a tree-structured filter bank. It has been widely explored in the literature [2, 20]. Here, the discussion is limited to general signal modeling issues.

The discrete wavelet transform is constructed by successive iterations on the lowpass branch. Given that $H_0(z)$ and $H_1(z)$ are respectively a lowpass and a highpass filter, the filtering operations can be readily interpreted. The first stage splits the signal into a highpass and lowpass band, each of which is spread to full band by the subsequent downsampling. Given this spreading that accompanies downsampling, the second stage can be viewed as simply splitting the lowpass portion of the original signal into halves. Each stage of the discrete wavelet transform thus splits the lowpass spectrum from the previous stage; this results in an octave-band decomposition of the signal, which is depicted in an ideal sense in 3.4.

As noted in the previous section, the deepest levels of iteration correspond to narrow frequency bands that necessarily lack time resolution. This tradeoff is very natural for octave-band decompositions. Low frequency signal components change slowly in time, so time resolution is not important. On the other hand, high frequency components are characterized by rapid time variations; to track such variations from period to period, for instance, time localization is important. This is exactly the time-frequency tradeoff provided by the discrete wavelet transform. Since the auditory system exhibits such frequency-dependent resolution, the wavelet approach has been considered for the application of auditory modeling [162, 163, 164].

The time-frequency localization in a given subband depends on its depth in the filter bank tree. A mathematical treatment of this is most easily carried out for a specific example. Consider a wavelet filter bank tree of depth three. By interchanging filters

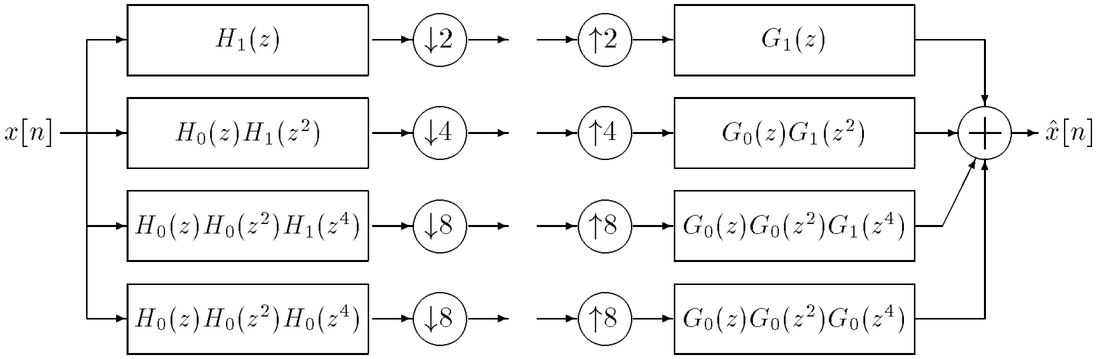

FIGURE 3.5: A tree-structured wavelet filter bank with three stages of iteration can be manipulated into this equivalent form.

and downsamplers in the analysis bank and interchanging filters and upsamplers in the synthesis bank, a depth-three discrete wavelet transform filter bank based on the filters $G_0(z), G_1(z), H_0(z)$, and $H_1(z)$ can be recast into the form shown in Figure 3.5; here, the deepest branches of the wavelet tree are now the filters with the most multiplicative components and the highest downsampling factors. The frequency-domain multiplication serves to narrow the frequency response and improve the frequency localization; the corresponding time-domain convolution serves to broaden the impulse response and decrease the time resolution. This spreading is shown in Figure 3.6 for a type of Daubechies wavelet that will be used for all of the wavelet-based simulations in this thesis; the functions shown are the impulse responses of the synthesis filters in Figure 3.5. Note that the subband signals in the wavelet filter bank are at different sampling rates; appropriately, the narrowest bands have the lowest sampling rate. Furthermore, it is important to keep in mind that the synthesis filter bank is required for aliasing cancellation.

### Atoms and filters

Earlier, the atomic model of the subband signals in a two-channel filter bank was derived. A similar model can be arrived at for the discrete wavelet transform [2]. The transform can thus be interpreted as a filter bank or as an atomic decomposition; there is a similar duality here as in the interpretations of the STFT discussed in Section 2.2.1, and the interpretations are connected by way of the tiling diagram. The two interpretations are further linked by a notion of evolution in that a subband signal is derived as an accumulation of atoms corresponding to the impulse responses of the synthesis filter in that band. The evolution, however, is not signal-adaptive as in the sinusoidal model.

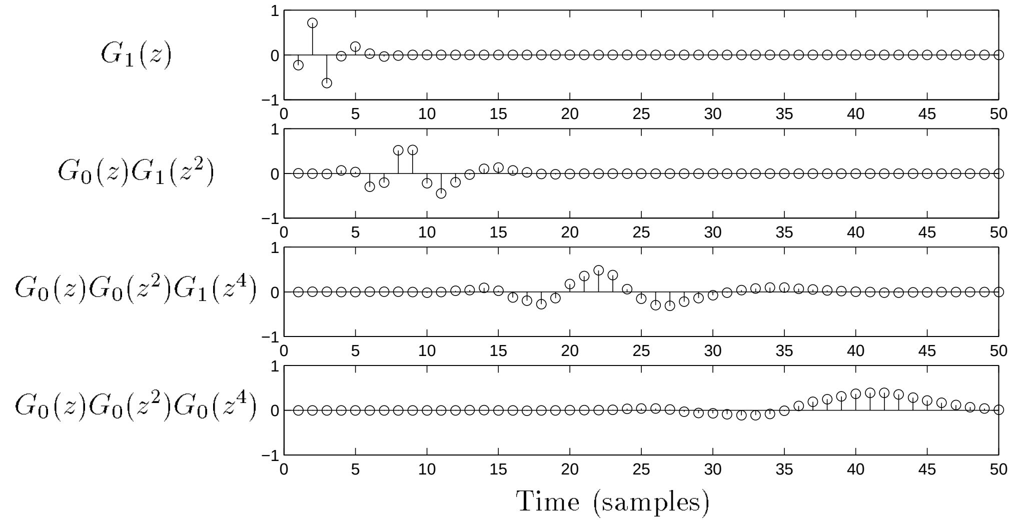

FIGURE 3.6: Impulse responses of a wavelet synthesis filter bank for a type of Daubechies wavelet. The expansion functions in the corresponding wavelet decomposition are these impulse responses and their shifts by 2, 4, 8, and 8 as indicated by the downsampling and upsampling factors in the filter bank of Figure 3.5.

### 3.2.2 Pyramids

Multiresolution decompositions can be derived using pyramid structures such as the one in Figure 3.7. These were originally introduced for multiresolution image processing [165]; the relationship to wavelets was realized shortly thereafter. The decomposition is again based on the idea of successive refinement; the signal is modeled as a sum of a coarse version (the top of the pyramid) plus detail signals.

There are several interesting things to note about the pyramid approach. Most importantly, perfect reconstruction is immediate; there are no elaborate constraints. This ease of perfect reconstruction is related to the fact that the pyramid decomposition is not critically sampled. Note that the coarse signal estimate derived at the highest level of the pyramid is analogous to the output of the lowest branch of a wavelet filter bank tree, but that the detail signals in the pyramid scheme are at higher rates than the corresponding detail signals in a wavelet filter bank; the output signal at the lowest level of the pyramid is itself full rate. For the pyramid in Figure 3.7, the representation is oversampled by a factor of $1 + \frac{1}{2} + \frac{1}{4} = \frac{7}{4}$; for continued iterations, the oversampling factor asymptotically approaches two. Along with simplifying perfect reconstruction, this oversampling results in added robustness to quantization noise [2]. Note also that the synthesis filters are included in the analysis; the result is an analysis-by-synthesis process that can be made to resolve some of the difficulties in wavelet filter banks. For instance, a pyramid-structured filter bank can be defined such that the subband signals are free of aliasing [166, 167].

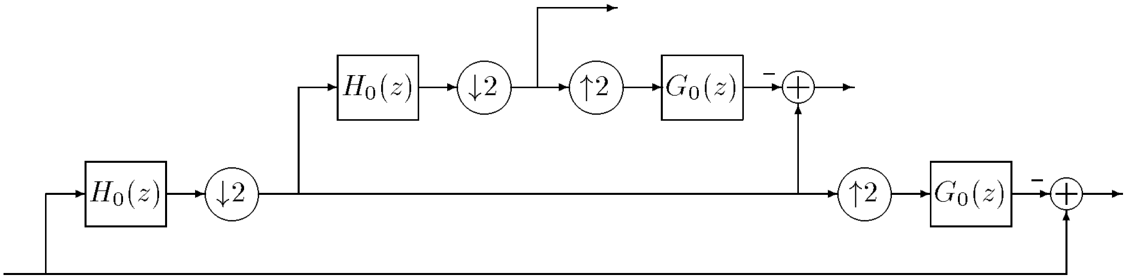

FIGURE 3.7: A pyramid structure for multiresolution filtering. This diagram depicts the analysis filter bank of the pyramid approach, which actually incorporates the synthesis process to ensure perfect reconstruction; synthesis is carried out by a structure similar to the right side of the analysis pyramid.

In the depiction of Figure 3.7, the signal decomposition is based on successive applications of the same filter pair $\{H_0(z), G_0(z)\}$. This is just one specific example of a pyramid approach, however. The pyramid structure can be generalized by applying arbitrary signal models on the levels of the pyramid rather than filtering and downsampling; for instance, in image coding it is common to apply nonlinear interpolation and decimation operators in such pyramid filters [2].

## 3.3 Filter Bank Methods

Filter bank methods for multiresolution sinusoidal modeling involve modeling the subband signals; a basic block diagram for this subband approach is given in Figure 3.8. The signal is split into bands of varying width, and each subband signal is the modeled with a separate sinusoidal model with resolution commensurate to the bandwidth – for narrow bands, long windows are used, and for wide bands, short windows are used. The filter bank in Figure 3.8 is shown as a generalized block since it may take the form of a discrete wavelet transform, an adaptive wavelet packet, a pyramid structure, or a nonsubsampled filter bank. These are discussed in turn in the following sections. Noting the similarity of this structure to that of Figure 2.6, these methods based on filter banks can be interpreted in some sense as multiresolution phase vocoders.

Note that the methods to be discussed generally involve octave-band filtering, which is perceptually reasonable since the auditory system exhibits roughly constant-$Q$ resolution [162]. Such octave-band filtering is useful with regards to the pre-echo problem. As shown in Section 2.6, the pre-echo depends on the window length; by using smaller windows for higher frequencies, the pre-echo becomes proportional to frequency in these filter bank methods. This proportionality is psychoacoustically viable in that perception

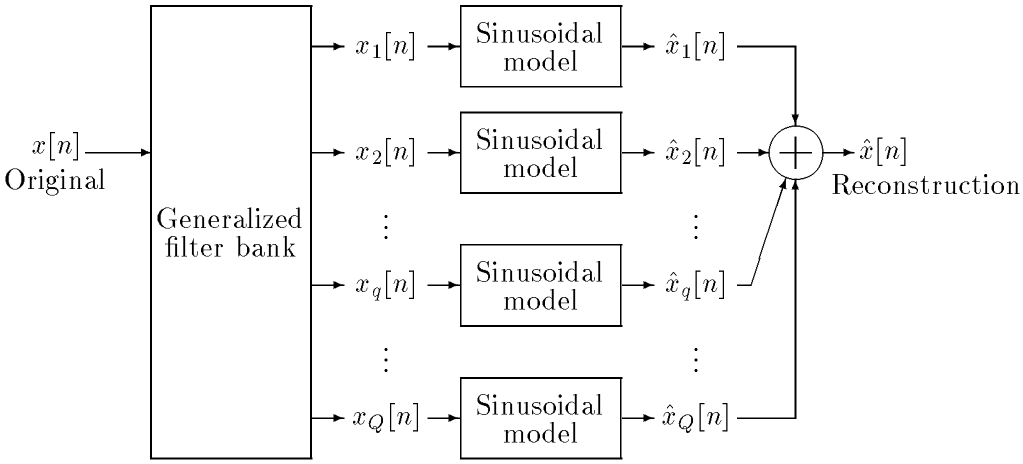

FIGURE 3.8: General structure of subband sinusoidal modeling. Alternatively, the sinusoidal model can be designed to yield signals that are intended as inputs to a synthesis filter bank, but this method has difficulties with aliasing cancellation.

of pre-echo is seemingly dependent on frequency; for a given partial, the percept depends not on the absolute length of the pre-echo but rather on how many periods of the partial occur in the pre-echo [168]. With that principle in mind, it is clear that pre-echo distortion can be alleviated by using long frames for low-frequency partials and short frames for high-frequency partials.

### 3.3.1 Multirate Schemes: Wavelets and Pyramids

Multirate systems are effective for dividing signals into subbands with low complexity and, in the critically sampled case, without increasing the amount of data in the representation. However, the analysis filtering process generally introduces aliasing, so the synthesis must incorporate aliasing cancellation to achieve a reasonable signal reconstruction. This aliasing leads to difficulties in the wavelet case that can be resolved by using a pyramid structure [168]; in Section 3.3.2, such issues are circumvented by using a nonsubsampled filter bank.

#### Wavelets

Sinusoidal modeling based on wavelet filter banks can be carried out in several ways. One approach is to model the downsampled subband signals, carry out a sinusoidal reconstruction of each subband at the downsampled rate, and use a wavelet synthesis filter bank to construct the full-rate signal. The same frame length is used in each subband. Then, because the lowpass band has the lowest sample rate, the lowpass frames have the longest effective time support; similarly, the frames in the highpass band have the shortest time support. This modeling method thus results in a parametric signal representation with the multiresolution properties of the discrete wavelet transform. As noted in [169], however, this method has difficulties because the sinusoidal model does not provide perfect reconstruction; aliasing cancellation is not guaranteed in the synthesis filter bank because the subbands are modified in the modeling process. This difficulty can be circumvented by reconstructing the output from the subband models without using the synthesis filter bank; the full rate reconstruction is derived directly from the models of the downsampled subbands [169]. In this method, it is necessary to explicitly account for aliasing in the sinusoidal parameter estimation; aliasing cancellation is incorporated into the estimation of the subband spectral peaks, but this typically accounts for only the aliasing between adjacent bands [170]. This method has reportedly proven useful for speech coding and time-scaling [169, 170]. An earlier hybrid algorithm involving wavelet-like filtering and sinusoidal subband modeling was reported in [171] for the application of source separation; here, the filter bank is oversampled in order to reduce the aliasing limitations.

#### Wavelet packets

In the approaches discussed above, the subbands of a wavelet filter bank are represented with the sinusoidal model to allow for modifications and processing. Such techniques can be conceptually generalized to the case of adaptive wavelet packets, where the tree-structured filter bank is varied in time according to the signal behavior; heuristically, the adaptation can be interpreted as follows: during transient behavior, the filter bank is characterized by short impulse responses to track the time-domain changes, and during stationary behavior the impulse responses are lengthened to improve the frequency resolution. Such wavelet packet vocoders have not been formally considered in the literature.

#### Pyramid structures

Octave-band filtering without subband aliasing can be carried out using a pyramid structure [166]. As in the pyramid structure of Figure 3.7, the subband representation is oversampled by a factor of two (asymptotically); here, the overcomplete representation provides an improvement over the critically sampled case in that the subbands are free of aliasing. This filter bank has recently been proposed as a front end for multiresolution sinusoidal modeling. The resulting algorithm has been shown to be effective for modeling a wide range of audio signals [168].

### 3.3.2 Nonsubsampled Filter Banks

In multirate filter banks, perfect reconstruction is achieved through the process of aliasing cancellation. In other words, there is inherently some degree of aliasing in the subband signals that is cancelled by the synthesis filter bank. This cancellation is a very exacting process; if an approximate representation such as the sinusoidal model is applied in the subbands prior to synthesis, aliasing cancellation in the reconstruction is not guaranteed.

The methods discussed above use various approaches to overcome aliasing problems. These issues never arise, however, if a nonsubsampled filter bank is used to split the input signals into the requisite bands. Such filter banks satisfy the perfect reconstruction constraint

$$\sum_q x_q[n] \;=\; x[n], \tag{3.13}$$

meaning that there is no aliasing or distortion introduced in the subband signals. The design of nonsubsampled filter banks that meet this constraint is very straightforward; the design process is discussed explicitly in Section 4.3.1. A decomposition in terms of alias-free subbands that meet the condition given in Equation (3.13) can indeed be arrived at using a nonsubsampled wavelet filter bank; the design method in Section 4.3.1, however, allows for more flexible spectral decompositions than the octave-band model derived by a wavelet filter bank.

In the multirate filter banks previously discussed, the subbands have different sampling rates. Then, a window of some fixed length can be applied in the subbands; with respect to the original sampling rate, the window in the lowpass band has the longest time support and the window in the highpass band has the shortest time support. The multiresolution in that case is provided by the multiplicity of sample rates. In the case of a nonsubsampled filter bank, multiresolution is achieved by using windows of different lengths in the subbands. This approach is depicted in a heuristic sense in Figure 3.9 for the case of a nonsubsampled octave-band filter bank.

Nonsubsampled filter banks are subject to much looser design constraints than multirate filter banks; this advantage arises because no aliasing cancellation is required. However, nonsubsampled filter banks have a seeming disadvantage with respect to multirate structures in that more computation is required to perform the filtering. Furthermore, in the nonsubsampled filter banks designed according to the method of Section 4.3.1, all of the filters in the filter bank are required to be of the same length; this supports the contention that multirate structures are more appropriate for multiresolution analysis. However, this is a somewhat inappropriate conclusion for the application at hand; as long as the filter bank impulse responses are of shorter duration than the sinusoidal analysis windows, the time resolution is limited by the subband sinusoidal models and not by the filter bank. Again, note that in the multirate structures the same window and stride can be used in each of the subbands; the multiresolution in those cases results from the fact that the subbands have different sampling rates. In nonsubsampled filter banks, multiresolution is achieved by choosing different window sizes and strides in the various subbands.

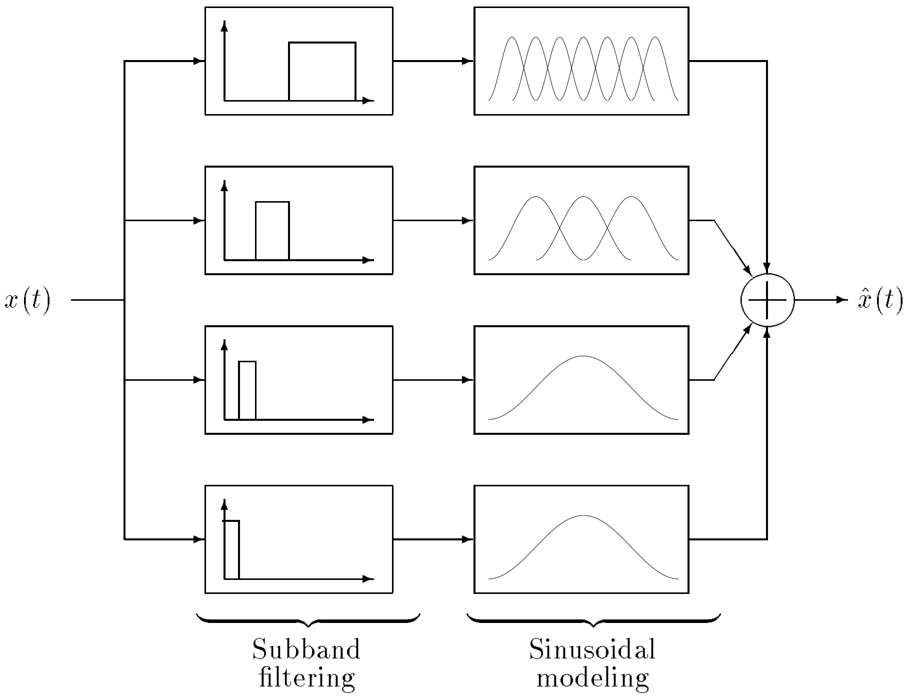

Figure 3.9: Multiresolution sinusoidal modeling with a nonsubsampled filter bank. The filter bank in this simple depiction provides an octave-band decomposition; the sinusoidal models have frame sizes scaled by powers of two according to the width of the respective subband. As described in the text, it is straightforward to design filter banks that derive other decompositions but it is not feasible to optimize the filter bank and the sinusoidal models for modeling arbitrary signals.

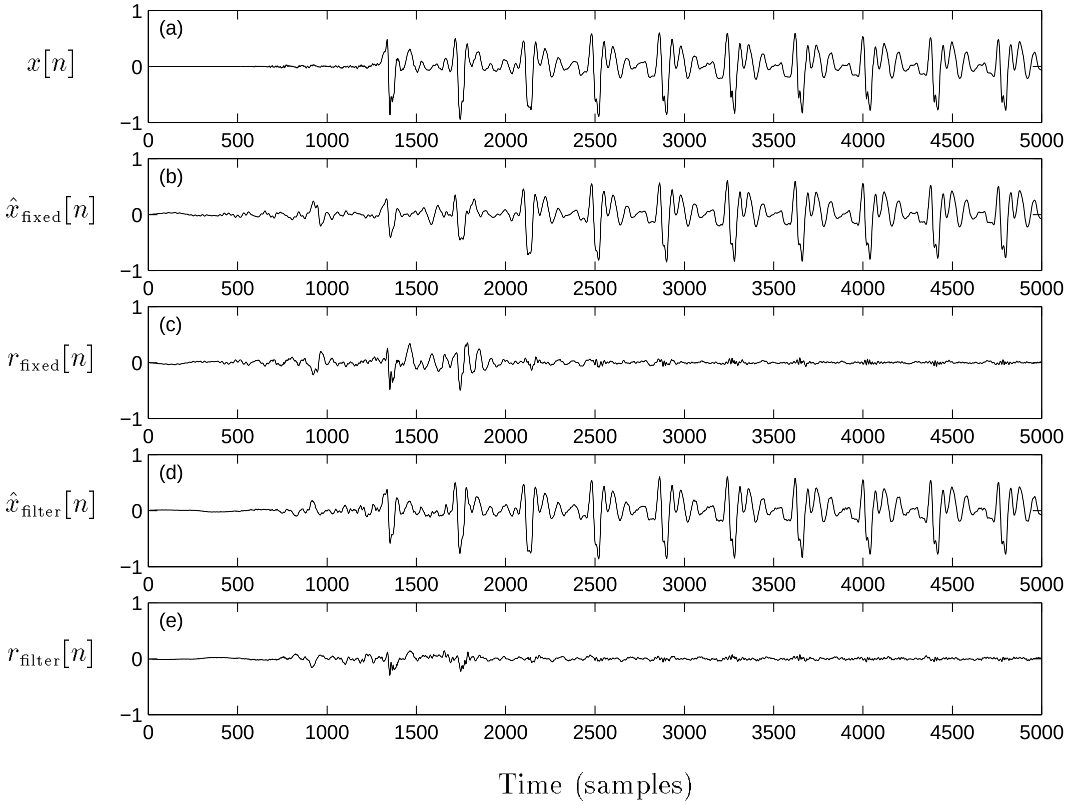

Figure 3.10: Multirate sinusoidal modeling using a nonsubsampled filter bank. The original signal in (a) is the onset of a saxophone note. Plot (b) is a sinusoidal reconstruction using a fixed frame size of 1024; plot (c) is the residual for that case. The plot in (d) shows a reconstruction based on sinusoidal modeling of the subbands of a nonsubsampled 7-band octave filter bank. Ranging from the lowest to the highest band, the subband sinusoidal models use synthesis frame sizes of 1024, 768, 512, 512, 256, 256, and 256. Plot (e) shows the residual for the filter bank case.

For a multiresolution sinusoidal model based on a filter bank, optimal design is prohibited by the large number of design parameters. The performance is influenced in complicated ways by the choices of filter band edges and frequency response properties as well as the parameters of the subband sinusoidal models (the number of partials, the window sizes, and the analysis strides). While heuristic designs can lead to modeling improvements as shown in Figure 3.10, a given design is not necessarily ideal for arbitrary signals. In a sense, if the filters and subband models are fixed, the problem is again a lack of signal adaptivity; the approach is rigid and can thus break down for some signals. In the next section, a signal-adaptive multiresolution framework based on time segmentation is considered.

## 3.4 Adaptive Time Segmentation

This section considers algorithms for deriving signal models based on adaptive time segmentation. The idea is to allow segments of variable size in a model so that appropriate time-frequency localization tradeoffs can be applied in various regions of the signal. Such a signal-adaptive segmentation can be arrived at by an exhaustive global search, by a dynamic program, or by a heuristic approach. These three methods are discussed in this section; the focus is placed on dynamic programs for segmentation, which can arrive at optimal models with substantially less computation than a global search.

### 3.4.1 Dynamic Segmentation

Given an entire signal and arbitrary allowances for intensive off-line computation, an optimal segmentation with respect to some modeling metric can be derived by a globally exhaustive search. If the metric is additive and independent across segments, however, the computational cost can be substantially reduced using a dynamic program. This approach has been applied to wavelet packet and LPC models [41, 60, 134]; after a brief review of dynamic programming and the relevant literature, dynamic segmentation for sinusoidal modeling is considered.

#### Dynamic programming

Dynamic programming was first introduced for solving minimum path-length problems [172]. The notion is that the computational cost of some classes of problems can be reduced by solving the problems in sequential stages; redundant computation is avoided by phrasing a global decision in terms of successive local decisions. This type of approach has found widespread use for sequence detection in digital communication, where it is referred to as the Viterbi algorithm [124]. Similar ideas play a role in hidden Markov modeling, which is central to many speech recognition systems [173, 174].

The dynamic programming method can be outlined as follows [175]:

- Consider the choice of a solution as a sequence of decisions.
- Incorporate a metric for the decisions such that the metric for the overall solution is the sum of the metrics for the individual sequential decisions.
- Assuming that a subset of the necessary decisions has been made, determine which decisions must be considered next and evaluate the metric for those decisions.
- Starting at the point where no decisions have been made, carry out a recursion to determine the set of decisions that are optimal according to the additive metric.

This description is rather general since the dynamic programming approach is itself quite general. The issues at hand are further clarified in the following discussion of the application of dynamic programming to signal segmentation and modeling; also, the computational efficiency afforded by dynamic programming will be quantified.

### Notation and problem statement

A mathematical treatment of the segmentation problem requires the introduction of some new notation; this is given here along with various assumptions about the signal and the computation requirements for modeling. First, there is some smallest segment size $\epsilon$ for the signal segmentation. Segments of length $\epsilon$ will be referred to as *cells*, and it will be assumed that the signal is $N$ cells long, *i.e.* the signal is of length $N\epsilon$. For general signal modeling, it is of interest to have a very flexible set of segment lengths to choose from; the set, which will be denoted by $\Lambda$, is thus assumed to consist of consecutive integer multiples of the cell size:

$$\Lambda \ = \ \{\epsilon, 2\epsilon, 3\epsilon, \ldots, L\epsilon\}. \tag{3.14}$$

A particular element from such a set of segment lengths will be denoted by $\lambda$.

Two specific cases will be considered in the treatment of computational cost. The first case is $L = N$, which implies that the implementation has no memory restrictions; for a signal of arbitrary length, the algorithm is capable of computing a model on a segment covering the span of the entire signal. The second case is $L < N$ (and sometimes $L << N$), which corresponds to the case of an implementation with finite memory. This restriction on $L$ is somewhat analogous to the *truncation depth* commonly used to reduce the delay in Viterbi sequence detection [124].

Using a diverse set of segment lengths allows for flexibility in signal modeling. Additional signal adaptivity can be achieved by allowing for a choice of model for each segment. One example of such a model choice is the filter order in an LPC application [134]. In the sinusoidal modeling case to be discussed, there is not a multiplicity of candidate models for each segment. For this reason, model multiplicity is not considered here. This omission is further justified in that if the evaluations of each model on a given segment require the same amount of computation, allowing for a choice of model does not affect the computation comparisons to be given.

The problem of signal modeling with adaptive segmentation is simply that of choosing an appropriate set of disjoint segments that cover the signal. The segmentation is chosen so as to optimize some metric; for proper operation of the dynamic program, it is required that the metric be independent and additive on disjoint segments. Then, the total metric for a segmentation $\sigma$ composed of segments $\lambda_i$ can be expressed as a sum of the metrics on the constituent segments:

$$D(\sigma) \ = \ \sum_i D(\lambda_i), \tag{3.15}$$

where $i$ is a segment index and where the constituent disjoint segments of the segmentation $\sigma$ satisfy

$$N \ = \ \sum_i \lambda_i \tag{3.16}$$

for a signal of length $N\epsilon$. Mean-squared error and rate-distortion metrics can be applied in this framework [41, 60, 134].

### Computational cost of global search

The globally optimum segmentation is simply the segmentation which minimizes the metric $D(\sigma)$. Obviously, this minimization can be arrived at by a globally exhaustive search in which the metric is computed for every possible segmentation in turn. The brief consideration here indicates that this exhaustive approach is computationally prohibitive for long signals; this difficulty motivates formulating the metric computation as a dynamic program.

In a globally exhaustive search, a model must be evaluated on each segment in each possible segmentation. Assuming that the cost of model evaluation is independent and additive on disjoint segments, a simple estimate of the computational cost of a global search can be arrived at by counting the total number of segments in all of the possible segmentations. This measure assumes that the cost of model evaluation on a segment is independent of the segment length. This assumption is admittedly somewhat unrealistic; for example, an FFT of a segment of length $\lambda$ requires on the order of $\lambda \log \lambda$ multiplies. The computational cost for other types of models are generally dependent on the segment length as well, so this enumeration of segments is by no means a formal cost measure but rather a basic feasibility indicator.

Given the discussion above, it is simply of interest to count the total number of segments in all of the possible segmentations. For the case $L = N$, this enumeration can be derived by simple combinatorics. Noting that there are $N-1$ cell boundaries in the interior of the signal and that each of these can be independently chosen as a segment boundary in the signal segmentation, there are $2^{N-1}$ possible segmentations; furthermore, the average number of segments in a segmentation is $(N+1)/2$. The total number of segments in all of the possible segmentations is given by

$$\mathcal{C} \;=\; [\text{number of segmentations}]\,[\text{number of segments per segmentation}], \qquad (3.17)$$

so the cost of global search for the case $L = N$ is

$$\mathcal{C}_{L=N} \;=\; 2^{N-2}(N+1), \qquad (3.18)$$

which is governed by an exponential dependence on the signal length:

$$\mathcal{C}_{L=N} \;\propto\; 2^{N}. \qquad (3.19)$$

In the truncated case $L < N$, the segment count does not have a simple formulation as in the unrestricted case. It can be shown, however, that the total number of segments is still governed by an exponential dependence on the signal length.[1] In either of these cases, the exponential dependence on the signal length prohibits model evaluation via exhaustive computation.

The next section describes a dynamic program that can derive the same optimal segmentation as an exhaustive search, but with a cost that is governed by a quadratic dependence on the signal length for the case $L = N$ and a linear dependence for $L << N$. As will be seen, this cost reduction is achieved by removing redundant computation; the simple insight in dynamic programming is that though some segment $\lambda$ is a component of many distinct segmentations, it is not necessary to calculate $D(\lambda)$ for each such occurrence. A dynamic program provides a computational framework in which $D(\lambda)$ is only evaluated once and hence the cost of evaluating a model on $\lambda$ is incurred only once.

### Reduction of computational cost via dynamic programming

The first step in a dynamic approach to signal segmentation is to consider the time span of the signal as a concatenation of cells. The boundaries between cells will be referred to as *markers*; because of the integer-multiple construction of the allowable segment lengths, the boundaries in any valid segmentation will align with some of these markers, so they can effectively be used as indices. In the dynamic program, each marker is treated as a possible segment boundary for the signal segmentation.

[1] In the case $L < N$, the number of possible segmentations is given by the $N$-th term of an $L$-th order Fibonacci series; this $N$-th term has an exponential dependence on $N$. Following the framework of Equation (3.17), the total number of segments in all of the possible segmentations is then given roughly by the product of this exponential term and the signal length.

Without loss of generality, the algorithm will be explained in terms of the examples shown in Figures 3.11 and 3.12, which correspond to the cases $L = N$ and $L < N$, respectively. In the figures, $D_{ab}$ represents the distortion metric associated with the segment of length $(b - a)\epsilon$ between markers $a$ and $b$. Further notation required for the explanation is as follows. At any marker $a$, the dynamic algorithm has determined the segmentation that leads to the minimum distortion up to that marker. This partial segmentation will be denoted by $\sigma_a$ and the corresponding distortion will be denoted by $D(\sigma_a)$; this distortion is the minimum modeling metric achievable for segmenting the signal up to the $a$-th marker. The term $\lambda_a$ will be used to denote the length of the last segment in the segmentation $\sigma_a$ that achieves the minimum metric $D(\sigma_a)$; the algorithm keeps track of this value at each marker so that the optimal segmentation can be recovered by backtracking after the end of the signal is reached.

Using the notation established above, the steps of the algorithm in the case $L = N$ are as follows; this corresponds to the illustration in Figure 3.11:

- Evaluate $D_{01}$, the modeling metric for the cell between markers 0 and 1, and store the result as $D(\sigma_1)$.

- Evaluate $D_{12}$ and $D_{02}$.

- Find $D(\sigma_2) = \min\{D_{02}, D(\sigma_1) + D_{12}\}$. This minimum indicates the best segmentation $\sigma_2$ between markers 0 and 2.

- Store $D(\sigma_2)$ and $\lambda_2$, the length of the last segment in $\sigma_2$.

- Evaluate $D_{23}$, $D_{13}$, and $D_{03}$.

- Find $D(\sigma_3) = \min\{D_{03}, D_{13} + D(\sigma_1), D_{23} + D(\sigma_2)\}$. This minimum indicates the best segmentation $\sigma_3$ between markers 0 and 3.

- Store $D(\sigma_3)$ and $\lambda_3$.

- Evaluate $D_{34}$, $D_{24}$, $D_{14}$, and $D_{04}$.

- Find $D(\sigma_4) = \min\{D_{04}, D_{14} + D(\sigma_1), D_{24} + D(\sigma_2), D_{34} + D(\sigma_3)\}$.

- Store $D(\sigma_4)$ and $\lambda_4$.

- Continue in this manner until the end of the signal is reached; note that each successive marker introduces a larger number of new candidate segments for consideration. The minimum $D(\sigma_N)$ calculated at the last marker is the globally optimal metric; as mentioned earlier, the optimal segmentation $\sigma_N$ can be found by backtracking through the recorded segment lengths.

The last item in the above description suggests a noteworthy point. To determine the segmentation that yields the minimum metric, it is necessary to store the appropriate segment length at each marker. The minimum metric itself, however, can be computed without storing path information.

The computational cost of the algorithm described above, namely an enumeration of the number of segments on which models are evaluated, can be easily determined by considering Figure 3.11. The number of candidate segments that must be evaluated at each marker is equal to the value of the marker index, so the cost is simply

$$\begin{aligned}\bar{\mathcal{C}}_{L=N} &= 1+2+3+\ldots+N &(3.20)\\ &= \frac{1}{2}(N^2+N), &(3.21)\end{aligned}$$

where the bar is included in the notation $\bar{\mathcal{C}}$ to specify that the cost corresponds to a dynamic algorithm. Noting the dominant term in the above expression, the cost of a dynamic segmentation algorithm with $L=N$ can be summarized as:

$$\bar{\mathcal{C}}_{L=N} \propto N^2. \tag{3.22}$$

This quadratic dependence on the signal length is a considerable improvement over the exponential dependence of an exhaustive global search.

For the case $L<N$, depicted in Figure 3.12, the steps in the algorithm are the same as above, with the exception of the later stages where the bounded segment length comes into effect:

- Evaluate $D_{01}$ and store the result as $D(\sigma_1)$.
- Evaluate $D_{12}$ and $D_{02}$.
- Find $D(\sigma_2) = \min\{D_{02}, D(\sigma_1) + D_{12}\}$.
- Store $D(\sigma_2)$ and $\lambda_2$.
- Evaluate $D_{23}$, $D_{13}$, and $D_{03}$.
- Find $D(\sigma_3) = \min\{D_{03}, D_{13} + D(\sigma_1), D_{23} + D(\sigma_2)\}$.
- Store $D(\sigma_3)$ and $\lambda_3$.
- Evaluate $D_{34}$, $D_{24}$, and $D_{14}$.
- Find $D(\sigma_4) = \min\{D_{14} + D(\sigma_1), D_{24} + D(\sigma_2), D_{34} + D(\sigma_3)\}$.
- Store $D(\sigma_4)$ and $\lambda_4$.

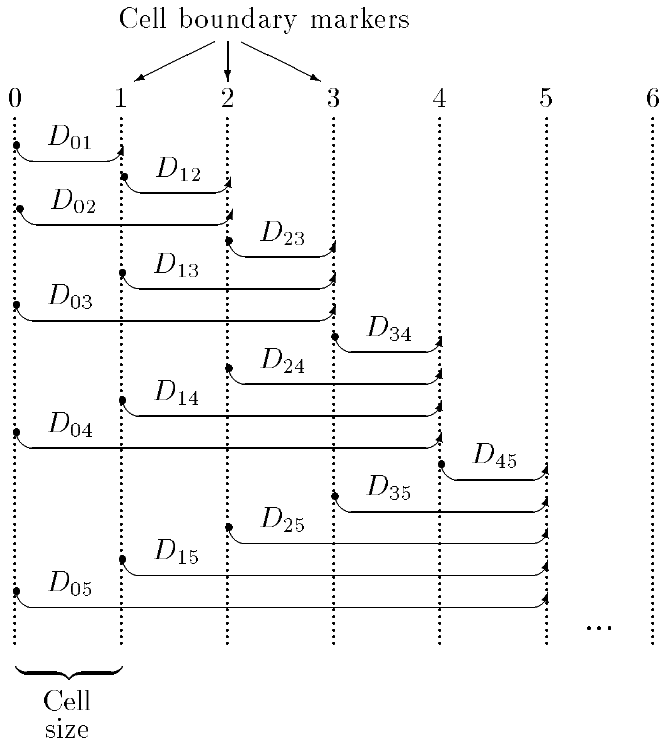

FIGURE 3.11: A depiction of a dynamic algorithm for signal segmentation for the case $L = N$, where the segment lengths are not restricted. As derived in the text, the computational cost of the algorithm grows quadratically with the length of the signal in this case.

- Continue in this manner until the end of the signal is reached; note that after marker $L$, each additional marker introduces a fixed number of candidate segments, namely $L$. The minimum $D(\sigma_N)$ calculated at the last marker is the globally optimal metric for this case; the optimal segmentation $\sigma_N$ can be found by backtracking through the recorded segment lengths.

The computational cost of the truncated approach can be readily derived by considering Figure 3.12, which indicates that the algorithm has a repetitive structure after the startup. The number of segments on which models are evaluated is given by

$$\begin{aligned} \bar{\mathcal{C}}_{L<N} &= \underbrace{1+2+\ldots+L-1}_{\text{startup}} + (N-L+1)L \qquad (3.23) \\ &= NL - \frac{1}{2}(L^2 - L). \qquad (3.24) \end{aligned}$$

The cost of a dynamic segmentation algorithm with $L < N$ can thus be summarized as

$$\bar{\mathcal{C}}_{L<N} \propto N, \qquad (3.25)$$

where the omission of the terms involving $L$ is particularly valid for cases where $L << N$, *i.e.* processing of arbitrarily long signals. For instance, in high-quality modeling of music it is necessary to have $L << N$ due to computational and memory limitations. Furthermore, it is sensible to restrict the segment lengths given that music is nonstationary in a global sense; it is unreasonable to assume that a one-segment model could describe an entire signal, so the candidate segment lengths can be justifiably bounded by some finite duration for which there is a possibility of local stationarity. In such cases, the cost grows linearly with the length of the signal, which is an improvement over both the global case of Equation (3.19) as well as the dynamic approach with unrestricted segment lengths described in Equation (3.22).

Applications of dynamic segmentation are discussed in the following; adaptive wavelet packets, linear predictive coding, and sinusoidal modeling can all be carried out in this framework. One caveat to note, however, is that in some of these methods it is necessary to use overlapping segments to ensure signal continuity at the synthesis frame boundaries. In such cases, the algorithm is not guaranteed to find the globally optimal segmentation; in practice, however, the effect is negligible, so the dynamic segmentations can be justifiably referred to as optimal [41]. A further issue to note is that the dynamic segmentation method, as described, considers the entire signal before a final decision is made regarding the segmentation; in this form, it is only suitable for off-line computation. In applications such as voice coding for telephony, it is of more interest to process the signal in blocks that can be transmitted sequentially. Dynamic segmentation can be applied in such scenarios by monitoring the candidate segmentation. The segmentations in signal regions that are distant in time are generally independent; without significantly

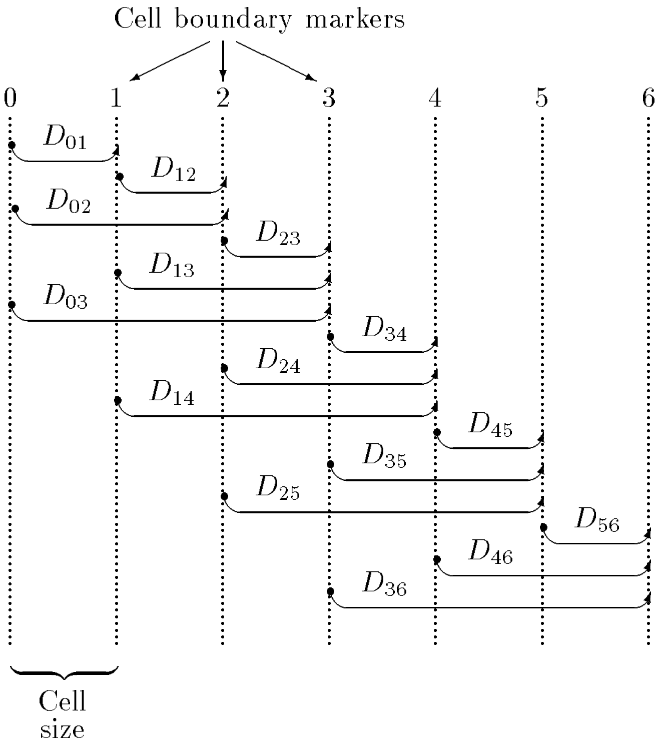

FIGURE 3.12: A depiction of a dynamic algorithm for signal segmentation for the case $L < N$, where the segment lengths are restricted. Note the regularity of the recursion after the startup; the cost of this algorithm grows linearly with the length of the signal.

sacrificing the optimality, then, the algorithm can be periodically terminated to derive blocks for coding [134].

### Adaptive wavelet packets

Early applications of dynamic programming to signal modeling involved models based on wavelet packets. In [41], the best wavelet packet in a rate-distortion sense is chosen for the model for each segment; in [60], dynamic segmentation is added to allow for localization of transients. A similar technique was considered in [176].

### Arbitrary models

In addition to the wavelet packet algorithms described, dynamic segmentation and model selection has been applied to image compression [177] and linear predictive coding [134]. As long as the optimality metric is independent and additive across disjoint frames, the dynamic program can be used to efficiently find the optimal segmentation and model selections. In cases where discontinuities across frame boundaries may be objectionable, the candidate models tend to have dependencies on adjacent frames; for instance, in the image processing application, where discontinuities result in blockiness, the candidate models are lapped orthogonal transforms which reduce the blocking artifacts incurred in the quantization [59, 177]. Because of the overlap, as mentioned before, the dynamic algorithm is not guaranteed to find the globally optimal model, but in practice the effect of the dependency is negligible. In the sinusoidal model application, as discussed below, the dynamic algorithm is again possibly suboptimal but this suboptimality turns out to be largely irrelevant.

### Sinusoidal modeling

As seen in Section 2.6, a sinusoidal model with a fixed frame size results in delocalization of time-domain transients if the frames are too long. This delocalization can be interpreted in terms of the synthesis: the signal is reconstructed in each synthesis frame as a sum of linear-amplitude, cubic-phase sinusoids, each of which has the same time support, namely the synthesis frame size; this fixed time support results in a smearing of signal features across the frame. In addition to this delocalization within each frame, features are spread across neighboring frames by the line tracking and parameter interpolation operations. One consequence of this is the pre-echo distortion discussed in Section 2.6; the example from Figure 2.21 is repeated here in Figure 3.14 for the sake of comparison with the improved models to be considered.

Time-domain delocalization, *e.g.* pre-echo distortion, results from the use of frames that are too long. If the frames are too short, a similar delocalization occurs in the frequency domain; frequency resolution is limited for short frames. For modeling arbitrary signals, then, it is of interest to trade off time and frequency resolution by selecting appropriate frame sizes, *i.e.* by deriving a dynamic segmentation of the signal based on an accuracy metric. Thus, in this application the metric $D(\lambda)$ is chosen to be the mean-squared error of the reconstruction over the segment $\lambda$; rate considerations can be easily incorporated by scaling the metric so as to favor longer frames, but this will not be dealt with here. In the implementation, the same number of partials is used in models of short frames and long frames to simplify the line tracking; because of this constant model order, using long frames improves the coding efficiency. In the simulations, the reduction of pre-echo is used as a visual indication of the modeling improvement. It will be clear that the algorithm chooses short frames to localize attacks, but it should also be noted that the method tends to choose longer frames when the signal exhibits stationary behavior, *i.e.* periodicity, since frequency resolution is increased in longer frames; this improved frequency resolution leads to more accurate modeling in periodic regions.

It was mentioned earlier that the dynamic algorithm is not guaranteed to find the optimal segmentation if the models in adjacent frames are dependent, but that such dependence is indeed required in some cases to prevent discontinuities in the synthesis. This scenario applies in the case of sinusoidal modeling. In the static case, the synthesis frames are demarcated by the centers of the analysis frames. There is thus an intrinsic overlap in the modeling process as depicted in Figure 3.13. This same overlap appears in the case of dynamic segmentation; as a result, the segmentation is not guaranteed to be optimal. The deviation from optimality, however, is basically negligible; the algorithm still carries out the intended task of finding appropriate tradeoffs in time and frequency resolution for modeling arbitrary signals. Figure 3.13 also indicates a noteworthy implementation issue, namely that a given segmentation requires a specific set of analysis windows to cover the signal. Each candidate segmentation thus has its own set of sinusoidal analysis results. These various analyses can be managed efficiently in the dynamic algorithm. Finally, it should be noted that the analysis windows, as depicted in Figure 3.13, need not satisfy the overlap-add condition. This design flexibility results from the incorporation of a parametric representation and applies to the general fixed-resolution sinusoidal model as well.

Fundamentally, the advantage of dynamic segmentation in the sinusoidal model is that the time support of the constituent linear-amplitude cubic-phase sinusoidal functions is adapted such that localized signal features are accurately represented. An example of the pre-echo reduction in such a multiresolution model is given in Figure 3.14. The dynamic algorithm chooses short frames near the attack to reduce delocalization, and long frames where the signal does not exhibit transient behavior.

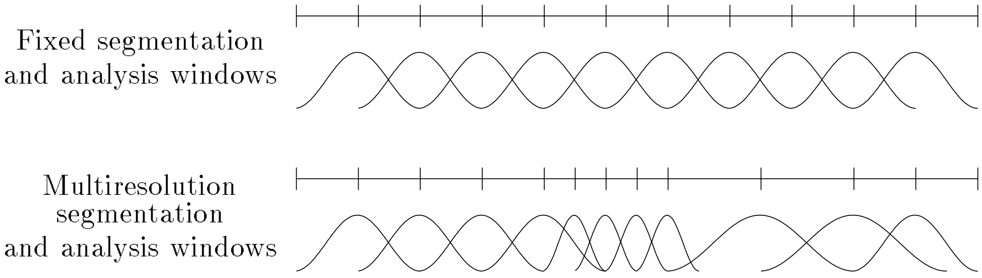

FIGURE 3.13: Analysis and synthesis frames in fixed-resolution and multiresolution sinusoidal models. This plot is included to indicate the overlap of the analysis frames. In the dynamic segmentation algorithm, this overlap undermines the required independence of the segment metrics; as a result, the synthesis segmentation derived by a dynamic program is not guaranteed to be globally optimal. This suboptimality is generally inconsequential, however.

### 3.4.2 Heuristic Segmentation

It is common in the development of signal processing algorithms to first investigate optimal or nearly optimal algorithms and then compare the results with lower cost methods based on less stringent metrics. In the framework of signal segmentation, this is tantamount to considering simple forward segmentation based on the immediate modeling error rather than focusing on global optimality. While the global segmentation is an analysis-by-synthesis approach that involves the entire signal, the forward segmentation is an analysis-by-synthesis that simply chooses among the candidate segments at each marker.

#### A simple algorithm for forward segmentation

In the sinusoidal model, a heuristic segmentation approach can achieve similar results as the dynamic algorithm for the example of Figure 3.14. The simple algorithm is as follows, where the signal segmentation is again described in terms of markers:

- At marker $a$, evaluate the weighted metric $\dfrac{D_{ab}}{b-a}$ for $b \in \{a+1, a+2, a+3, \ldots, a+L\}$, where the set corresponds to the candidate segment lengths.

- Find the marker $\hat{b}$ which minimizes the weighted metric and advance to that marker.

- Set a new starting point at $a = \hat{b}$ and repeat the preceding steps.

Note that in this algorithm the segmentation decisions are made based on local minimization of the distortion metric; since local minima are pursued greedily, global optimality of the metric is not guaranteed. Of course, many variations of forward segmentation can

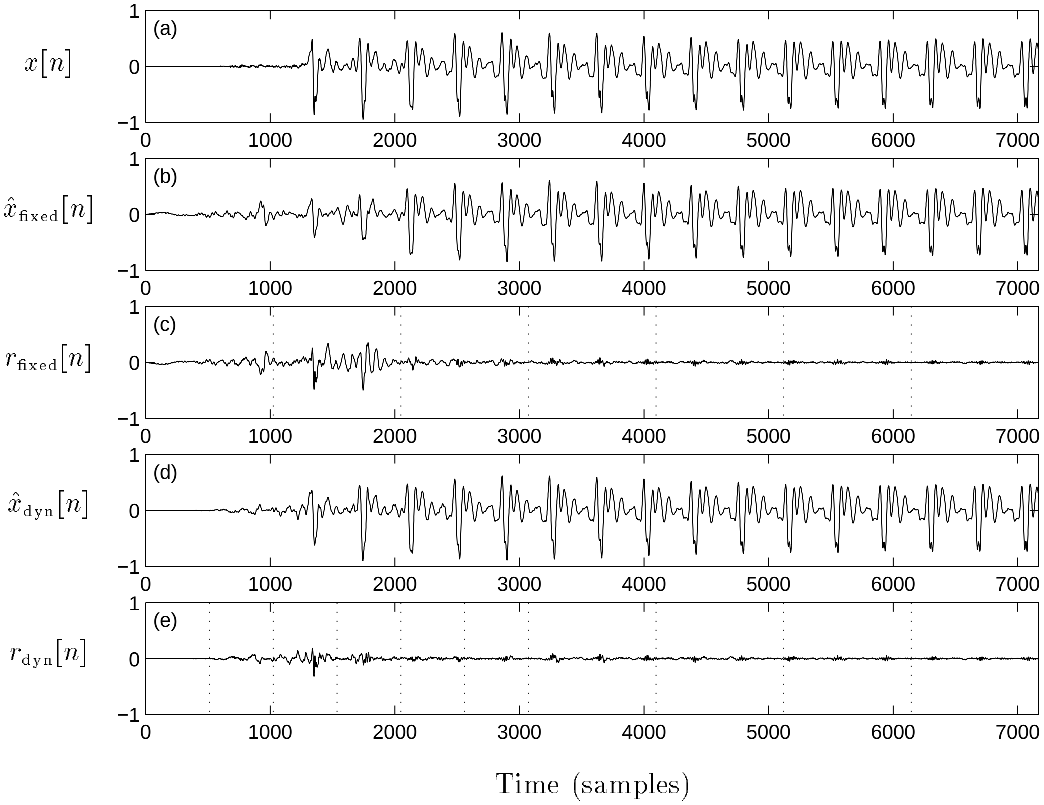

FIGURE 3.14: Comparison of residuals for a fixed-frame sinusoidal model and an adaptive multiresolution model based on dynamic segmentation. The original signal (a) is a saxophone note. Plot (b) is a reconstruction based on a fixed frame size of 1024 and (c) is the residual for that case; the dotted lines indicate the synthesis frame boundaries. Plot (d) is a reconstruction using dynamic segmentation with frame sizes 512, 1024, 1536, and 2048; the segmentation arrived at by the dynamic algorithm is indicated by the dotted lines in the plot of the residual (e). In the dynamic model, the attack is well-localized and does not contribute extensively to the residual (e).

be formulated; for instance, by incorporating some dependence on neighboring results, a more global solution can be targeted. Such variations will not be considered, however; the intent is merely to draw a comparison between dynamic and heuristic segmentation methods.

Figure 3.15 shows an application of forward segmentation to a saxophone attack; for this example, the forward method achieves a similar model as the dynamic algorithm, but such comparable performance is not guaranteed for all signals. As will be shown in the next section, the forward segmentation requires less computation than the dynamic approach. In real-time (or limited-time) applications, then, the reduced cost of a forward segmentation method may merit this accompanying decrease in modeling accuracy. On the other hand, in off-line applications such as compression of images or audio for databases, it is more appropriate to use an optimal dynamic algorithm.

### Cost of forward segmentation

In the heuristic segmentation algorithm described above, the number of markers visited depends on the signal; if a long frame is chosen, the algorithm advances to the end of the frame and skips over the markers in between. Thus, the computation required in the algorithm is signal-dependent. To quantify the computational cost, then, the worst case scenario is considered; the case in which every marker is visited provides an upper bound for the cost. For $L = N$, the number of segments considered at successive markers decreases as the algorithm advances toward the end of the signal; for the worst case, the cost is given by

$$\begin{aligned} \tilde{\mathcal{C}}_{L=N} &= N + (N-1) + (N-2) + \ldots + 2 + 1 && (3.26) \\ &= \frac{1}{2}(N^2 + N) && (3.27) \\ \Longrightarrow \quad \tilde{\mathcal{C}}_{L=N} &\propto N^2, && (3.28) \end{aligned}$$

where the tilde is included in the notation $\tilde{\mathcal{C}}$ to specify that the cost corresponds to a forward algorithm. For the case $L < N$, the worst case cost is given by

$$\begin{aligned} \tilde{\mathcal{C}}_{L<N} &= L + L + \ldots + L + \underbrace{L - 1 + L - 2 + \ldots + 2 + 1}_{\text{end of signal}} && (3.29) \\ &= NL - \frac{1}{2}(L^2 - L) && (3.30) \\ \Longrightarrow \quad \tilde{\mathcal{C}}_{L<N} &\propto N. && (3.31) \end{aligned}$$

The costs here are identical to those evaluated for the dynamic algorithm; compare Equations (3.27) and (3.30) with Equations (3.21) and (3.24). In either case, the total number of segments considered in the worst case forward segmentation is the same as the number

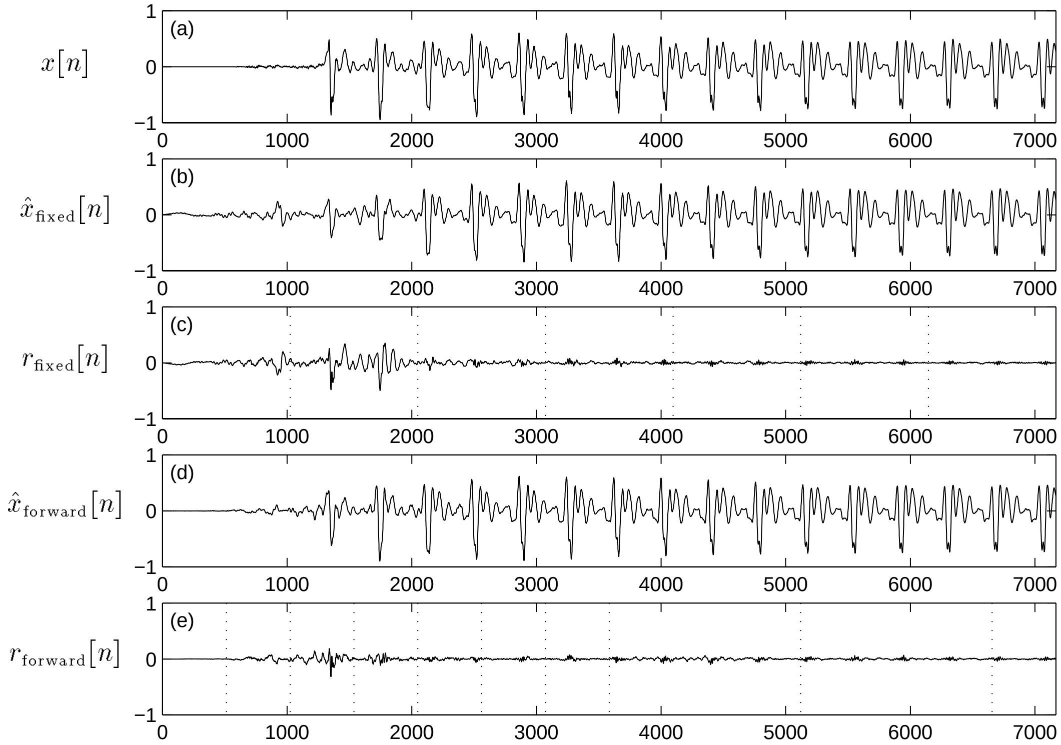

FIGURE 3.15: Comparison of residuals for a fixed-frame sinusoidal model and an adaptive multiresolution model based on forward segmentation. The original signal (a) is a saxophone note. Plot (b) is a reconstruction based on a fixed frame size of 1024 and (c) is the residual for that case; the dotted lines indicate the synthesis frame boundaries. Plot (d) is a reconstruction using forward segmentation with frame sizes 512, 1024, 1536, and 2048; the segmentation arrived at is indicated by the dotted lines in the plot of the residual (e). In the forward adaptive model, the attack is well-localized and does not contribute extensively to the residual.

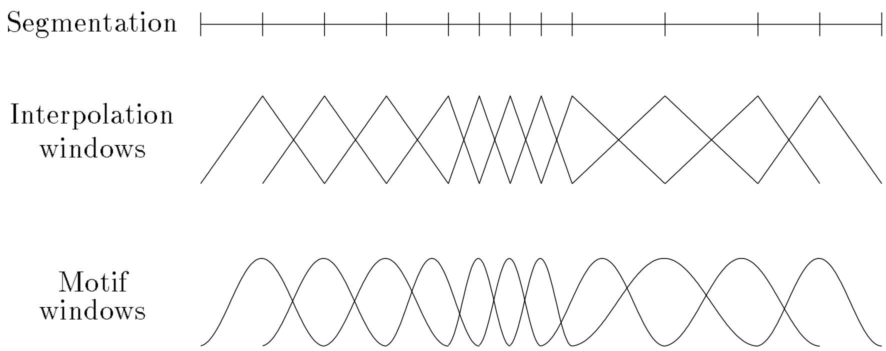

FIGURE 3.16: Multiresolution frequency-domain synthesis with dynamic segmentation involves symmetric motif windows and asymmetric interpolation and overlap-add windows.

considered in the dynamic algorithm. For the truncated case, a more optimistic formulation of the computation required in the forward approach can be arrived at by an averaging argument. Assuming that the segment lengths are all equally reasonable for modeling, and that the expected length of any given segment chosen by the algorithm is thus $(L+1)/2$, the forward algorithm is expected to visit only $2N/(L+1)$ markers. The cost estimate is then

$$\tilde{\mathcal{C}}_{L<N} = \frac{2NL}{L+1} \qquad (3.32)$$

$$\Longrightarrow \tilde{\mathcal{C}}_{L<N} \propto N, \qquad (3.33)$$

which has the same dependence on the signal length as the upper bound in Equation (3.31); noting the dependence on $L$ indicated in the formulation, however, it is clear that the average cost is roughly a factor of $L/2$ less than the worst case upper bound.

### 3.4.3 Overlap-Add Synthesis with Time-Varying Windows

The preceding discussion of dynamic segmentation in the sinusoidal model has focused on time-domain synthesis. For the sake of completeness, it is noted here that dynamic segmentation can also be applied in the frequency-domain synthesis approach discussed in Section 2.5. The fundamentals of such an approach are discussed below, and connections to current techniques in audio coding are described.

In the synthesizer described in Section 2.5, the signal is modeled in the frequency domain as a series of short-time spectra, from which the signal is reconstructed using an IFFT and overlap-add. Each of these short-time spectra is a sum of spectral *motifs* corresponding to short-time partials. The motif is basically the transform of some window function $b[n]$, so the IFFT results in a sum of sinusoids windowed by $b[n]$. The overlap-add is then carried out with the hybrid window $t[n]/b[n]$ where $t[n]$ is a triangular window which satisfies the overlap-add property. As described in Sections 2.5.1 and 2.5.2, this triangular OLA carries out reasonable interpolation of the sinusoidal parameters if phase matching is employed.

In a multiresolution implementation, it is necessary to incorporate motifs of various time resolution; for longer segment sizes, the short-time spectrum has more bins and the IFFT is larger. Recalling the discussion of Section 2.5, it is computationally important to use a symmetric motif window $b[n]$ and likewise a symmetric spectral motif. Adhering to this symmetry in a multiresolution setting results in asymmetric overlap-add windows; indeed, the interesting adjustment of the algorithm involves the overlap-add window and the effective interpolating window $t[n]$. Because of the variable segment sizes, to do the appropriate OLA interpolation it is necessary to use asymmetric triangular functions at transitions between different segment sizes. This approach is best described pictorially; Figure 3.16 shows a signal segmentation and the corresponding motif and interpolation windows. Note that the asymmetric transition windows are conceptually similar to the *start* and *stop* windows used in modern audio coding standards [7, 8]; in those methods, however, such asymmetric windows are used in conjunction with a filter bank analysis-synthesis and not with a parametric approach as in this consideration.

## 3.5 Conclusion

In modeling nonstationary signals, it is generally useful to carry out analysis-synthesis in a multiresolution framework; appropriate time-frequency resolution tradeoffs can be adaptively incorporated to achieve accurate compact models. In this chapter, the notion of multiresolution was introduced in terms of the discrete wavelet transform and further explored in the context of the sinusoidal model. Two methods of multiresolution sinusoidal modeling were discussed, namely filter bank techniques and adaptive time segmentation. A dynamic programming for signal segmentation was developed; related computation issues were considered at length. Various simulations in the chapter showed that multiresolution modeling improves the localization of transients in the sinusoidal reconstruction; this improvement was indicated by a mitigation of pre-echo distortion.

# Chapter 4

# Residual Modeling

The sinusoidal model, while providing a useful parametric representation for signal coding and modification, does not provide either perfect or perceptually lossless reconstruction for most natural signals. Thus, it is necessary to separately model the analysis-synthesis residual if high-quality synthesis is desired; this requirement was the motivation for the deterministic-plus-stochastic decomposition proposed in [36, 100]. This chapter discusses a parametric approach for perceptually modeling the noiselike residual for both time-domain and frequency-domain synthesis. Earlier versions of this work have been presented in the literature [110, 178].

## 4.1 Mixed Models

Mixed models have been applied in many signal processing algorithms. For instance, in linear predictive coding (LPC) of speech, the speech signal is typically classified as voiced or unvoiced to determine the synthesis model; in the voiced case, the synthesis filter is driven by a periodic impulse train whereas in the unvoiced case, the filter is driven by white noise. The model thus adapts to a nonstationary signal by choosing the appropriate excitation. In some variations of the algorithm, a mixed excitation is used to account for concurrent voiced and unvoiced signal behavior; using a mixture enables modeling of a wider variety of signals than with a switched excitation [25, 179]. The voiced-unvoiced model, especially in the case of a mixed excitation, is similar to the deterministic-plus-stochastic sinusoidal model decomposition proposed in [36, 100] and explored further in [97, 110, 178, 109, 180]. The components in these latter models are concurrent in time; the models are thus capable of representing a wide variety of signals.

In Section 2.1.2, where the deterministic-plus-stochastic decomposition was first described, it was noted that in the framework of analysis-synthesis it is natural to rephrase the decomposition in terms of a signal reconstruction and a residual. The reconstruction is based on the signal model, in this case the sinusoidal model; the residual is the difference

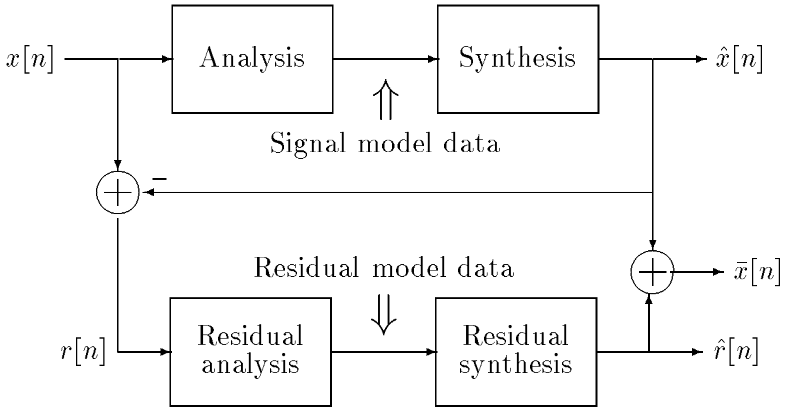

FIGURE 4.1: Analysis-synthesis and residual modeling.

between the original and the reconstruction. When the analysis-synthesis model does not capture all of the perceptually important features of a signal, it is necessary to separately model the residual and incorporate it into the reconstruction to achieve perceptual losslessness; this scenario, which applies in the case of sinusoidal modeling, is depicted in Figure 4.1. Such modeling of residuals is used in many audio applications as well as in other signal processing algorithms, for instance motion-compensated video coding [181]. These approaches are effective because the residuals tend to be "noiselike" – in some cases such as LPC, the signal model is indeed designed with the very intent of leaving a white noise residual. In modeling such noiselike residuals, it is important to account for perceptual phenomena. As discussed in Section 1.2.2, white noise processes are basically incompressible if perfect reconstruction is desired. On the other hand, compact models of noiselike residuals can readily achieve perceptual losslessness by incorporating simple principles of perception. Furthermore, it should be noted that the condition of transparency can be relaxed somewhat for the residual synthesis given the perceptual masking principles that come into effect when the modeled residual is recombined with the primary signal. The fundamental goal is for the recombination to be perceptually equivalent to the original signal, and not for the synthesized residual to be a transparent version of the original residual.

In music applications, the sinusoidal model captures the basic musical signal features such as the pitch and the spectral structure. The residual contains features that are not well-represented by the slowly-evolving sinusoids of the sum-of-partials model; as discussed in Sections 2.1.2 and 2.6, these correspond to musically important processes such as the breath noise of a flute or saxophone or the attacks of a piano or marimba. Multiresolution sinusoidal approaches were proposed in Chapter 3 to model the attacks, so the residual model of this chapter is designed to handle the remaining features, namely broadband stochastic processes such as breath noise. It is necessary to incorporate these processes into the reconstruction to achieve realistic or natural-sounding synthesis.

In [36, 100], the residual is modeled using a piecewise-linear spectral estimate; a random phase is applied to this spectrum, and an inverse discrete Fourier transform (IDFT) followed by overlap-add (OLA) is used for synthesis. In the approach to be discussed in this chapter, the model is similarly spectral in nature, but is more directly based on perceptual considerations. The residual is analyzed by a filter bank motivated by auditory perception of broadband noise; a parameterization provided by the short-time energy of the filter bank subbands yields a perceptually accurate reconstruction of the noiselike residual. Furthermore, the model parameters allow for modifications of the residual; this capability is useful in that if the sinusoidal signal components are modified, the residual should undergo a corresponding transformation prior to synthesis [142].

In [109, 180] the models are more elaborate than the one presented in this chapter in that they have specific extensions to model attack artifacts present in the residual, which were discussed in Section 2.6. The approach taken here is to use multiresolution sinusoidal modeling to minimize such artifacts so that they do not appear in the residual and thus do not have to be accounted for in the residual model. A similar approach is taken in the algorithm in [101], which estimates the time-domain envelope of the signal and applies it to the sinusoidal model to enhance the modeling of transients. This method involves incorporating another set of parameters to describe the time-domain envelope, however, so the multiresolution model has an advantage in that its representation is more uniform.

Figure 4.2 gives a comparison of the residuals for a basic sinusoidal model and multiresolution model based on dynamic time segmentation; more comparisons of this nature were given in Chapter 3. Clearly, the attack artifacts are not as pronounced in the residuals of the multiresolution model. Because of its improved ability to represent the signal transients, the residual energy in the dynamic model is lower; as discussed in Section 3.4, the multiresolution model is adapted to minimize this energy given various constraints such as the number of sinusoids in the model; the notion of minimizing the residual energy is also incorporated in the analysis-by-synthesis algorithm discussed in [101] and in global parameter optimization methods [107]. Also, in the methods to be discussed in later chapters, minimization of the residual energy is again the criterion by which the signal model is adapted.

## 4.2 Model of Noise Perception

Noting the example of Figure 4.2 and the results given in Chapter 3, it is assumed hereafter that attack transients have been well-modeled in a multiresolution framework. The residual thus consists of broadband noise processes. A perceptually viable model for the residual should therefore rely on a model of how the auditory system perceives

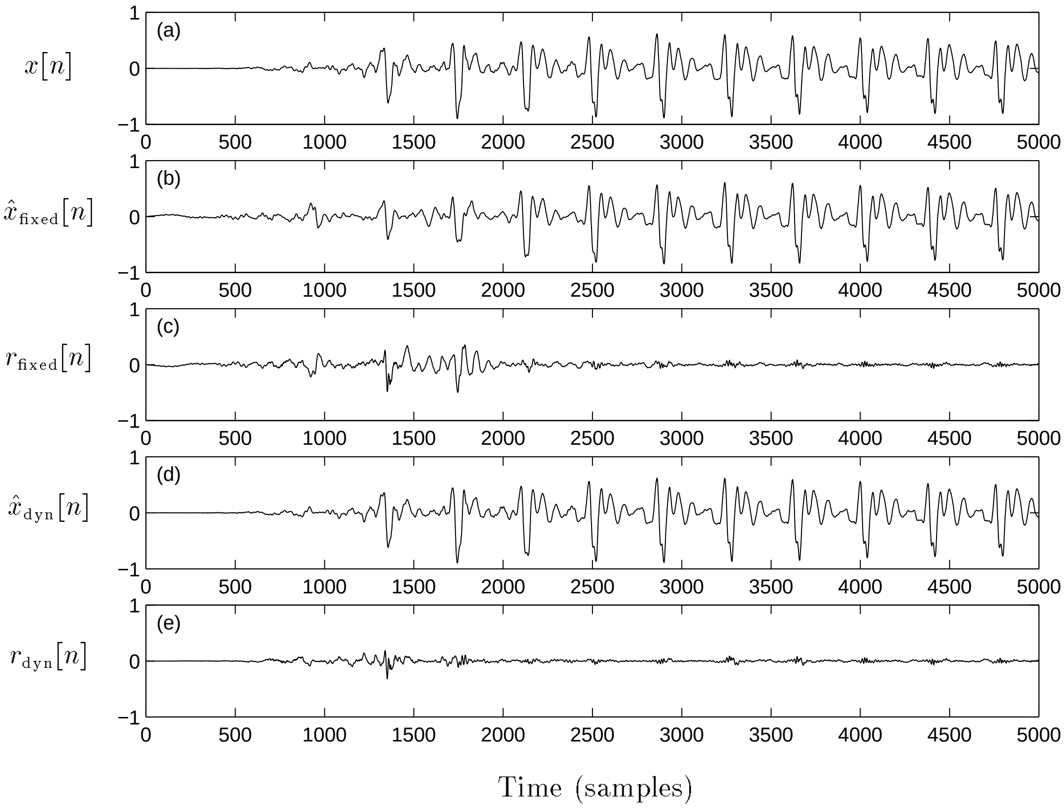

FIGURE 4.2: Comparison of residuals for fixed and multiresolution sinusoidal models. The original signal (a) is a saxophone note. Plot (b) is a reconstruction based on a fixed frame size of 1024 and (c) is the residual for that case. Plot (d) is a reconstruction using dynamic segmentation with frame sizes 512 and 1024; in this case, the attack is well-modeled and does not appear as extensively in the residual (e).

broadband noise. This section discusses a simple filter bank model of the auditory system that leads to a perceptually lossless representation of the residual.

### 4.2.1 Auditory Models

Auditory models commonly include a set of overlapping bandpass filters whose bandwidths increase roughly in proportion to their center frequencies. Such filter bank models, which were first introduced in conjunction with the classical theory of resonance [182], are well justified by experimental work ranging from early masking tests for telephony applications [183, 184] to recent investigations in perceptual audio coding, where auditory models are incorporated to achieve transparent compression [8, 7, 9, 10, 11] These auditory filter banks can be characterized in terms of the classical critical bandwidths, which were derived in experiments on noise masking and perception of complex sounds; these are generally considered to be the bandwidths of the auditory filters at certain center frequencies [185]. Early estimates of the critical bandwidth as a function of center frequency indicate a roughly constant value below 500 *Hz* and a linear increase for higher frequencies, resulting in the common interpretation of the auditory system as a constant-$Q$ filter bank. More recent experiments suggest that the low-frequency critical bandwidths are quadratically related to the center frequency [186]. Expressions for the *equivalent rectangular bandwidths* of the auditory filters differ somewhat from the bandwidth formulations in classical critical band theory; the difference is depicted in Figure 4.3. Of course, these results are based on aggregate measurements over large groups of subjects, so the exact relation does not necessarily apply to any given individual. Furthermore, for this application of residual modeling it is unnecessary to incorporate formal exactitudes about the auditory filter responses because the perception of broadband noise is an inherently coarse phenomenon. The purpose of the previous discussion, then, is only to support the notion of filter bank auditory models and to establish the terminology; for the remainder, an equivalent rectangular band will be referred to as an ERB.

### 4.2.2 Filter Bank Formulation

A simple model of noise perception can be arrived at by dividing the spectrum into a set of bands based on the ERB formulation. Given this division into bands, the basic model is that in perceiving a broadband noise, the auditory system is primarily sensitive to the total short-time energy in each of the bands, and not to the specific distribution of energy within any single band. In other words, the ear is insensitive to specific local time or frequency behavior of broadband noise. Analysis of a broadband noise $s[n]$, which corresponds to $r[n]$ in the residual modeling framework of Figure 4.1, is then carried out by first applying $s[n]$ to an ERB filter bank $\{h_1[n], h_2[n], \ldots, h_R[n]\}$ to derive the ERB signals $\{s_1[n], s_2[n], \ldots, s_R[n]\}$ as shown in Figure 4.4. These signals are then parameterized on a

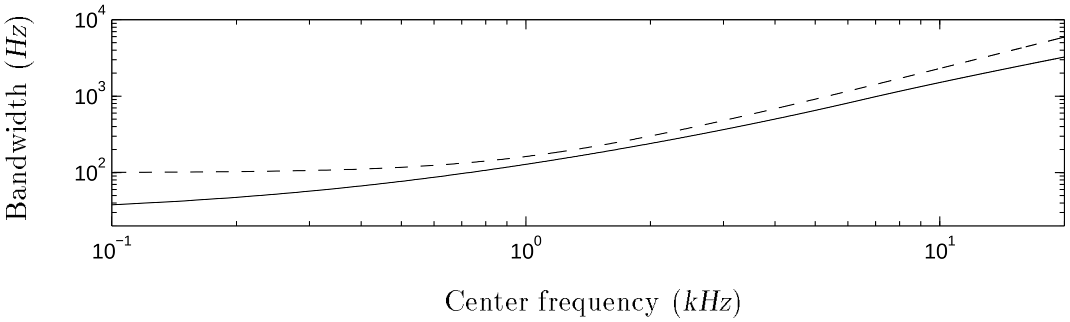

FIGURE 4.3: Bandwidth vs. center frequency for critical bands (dashed) and equivalent rectangular bands or *ERBs* (solid).

frame-rate basis in terms of their energies; for the $i$-th frame, the energy of the $r$-th ERB signal is given by

$$E_r(i) \;=\; \sum_{n=0}^{N-1} s_r[n+iL]^2, \tag{4.1}$$

where $N$ is the frame size and $L$ is the analysis stride. Synthesis according to this model is achieved by filtering white noise $\psi[n]$ through the ERB filter bank with a time-varying gain $c_r(i)$ on each channel; this structure is shown in figure 4.5.

The time-varying gains in the synthesis filter bank shape the short-time spectrum of the filter bank output $\hat{s}[n]$ so that it matches the short-time spectrum of $s[n]$ in the sense that their ERB energies are equivalent. The appropriate gain can be derived using a simple constraint on the expected value of the synthesis energy:

$$\mathrm{E}\{\hat{E}_r(i)\} \;=\; E_r(i). \tag{4.2}$$

Note that this filter bank model relies on the aggregation of filters only inasmuch as they span the signal spectrum; the interaction between filters is not important. The model is simply that the subband ERB signal $s_r[n]$ is perceptually equivalent to the subband reconstruction $\hat{s}_r[n]$ if their short-time energies meet the above constraint; then, if the filter bank is designed such that $s[n] \;=\; \sum_r s_r[n]$, perceptual losslessness holds for the entire filter bank model.

The appropriate gains can be derived by expanding the constraint of Equation (4.2). The expected value of the synthesis energy of the $r$-th band in the $i$-th frame is given by

$$\mathrm{E}\{\hat{E}_r(i)\} \;=\; \mathrm{E}\left\{\sum_{n=0}^{N-1} \left(c_r(i)\tilde{s}_r[n+iL]\right)^2\right\}, \tag{4.3}$$

where

$$\tilde{s}_r[n] \;=\; h_r[n] \;*\; \psi[n] \tag{4.4}$$

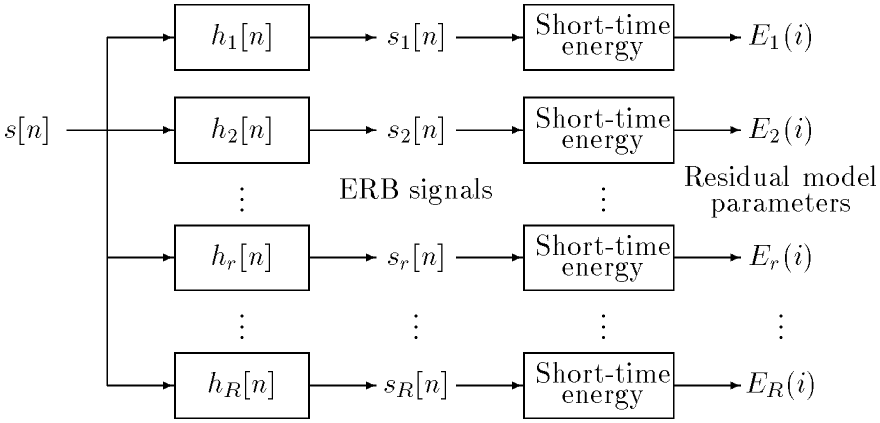

FIGURE 4.4: Analysis filter bank for perceptually modeling broadband noise. The residual is parameterized in terms of the short-time energies $E_r(i)$ in a set of equivalent rectangular bands (ERBs).

is the output of the $r$-th synthesis filter before the gain $c_r(i)$ is applied. Substituting this convolution into Equation (4.3) yields the following expression; the index $iL$ is dropped without loss of generality:

$$\mathrm{E}\{\hat{E}_r(i)\} = c_r(i)^2 \sum_{n=0}^{N-1} \mathrm{E}\left\{ \left( \sum_m h_r[m]\psi[n-m] \right)^2 \right\} \tag{4.5}$$

$$= c_r(i)^2 \sum_{n=0}^{N-1} \sum_m \sum_l h_r[m]h_r[l]\mathrm{E}\left\{\psi[n-m]\psi[n-l]\right\}. \tag{4.6}$$

Denoting the variance of the white noise $\psi[n]$ as $\sigma^2$, the expected value in the sum can be replaced by $\sigma^2\delta[m-l]$. Summing over $l$, the expression simplifies to:

$$\mathrm{E}\{\hat{E}_r(i)\} = c_r(i)^2 N\sigma^2 \sum_m h_r[m]^2. \tag{4.7}$$

Note that the filters have been assumed real so that the subband signals are real and thus immediately perceptually meaningful. Incorporating the constraint of Equation (4.2) provides a formula for the gain in terms of the ERB energy parameter:

$$c_r(i) = \sqrt{\frac{E_r(i)}{N\sigma^2 \sum_m h_r[m]^2}}. \tag{4.8}$$

Equation (4.8) can be interpreted in two ways. First, the appropriate gain $c_r(i)$ can be derived in the frequency domain as a ratio between the ERB energy in band $r$

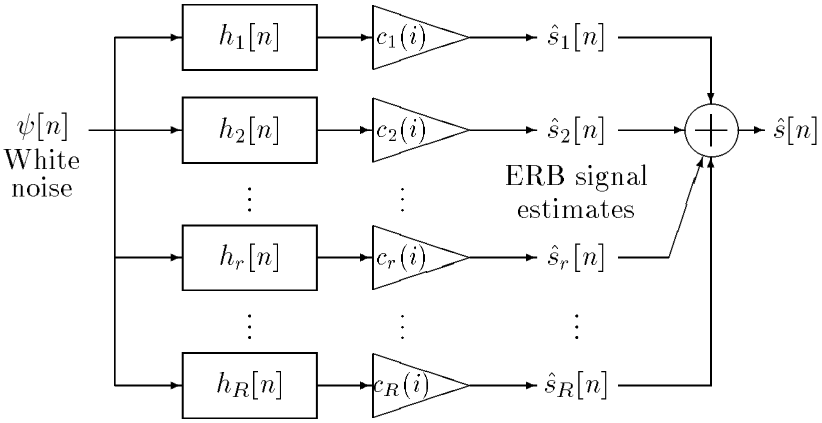

FIGURE 4.5: Synthesis filter bank for perceptually modeling broadband noise. The time-varying gains $c_r(i)$ given by Equation (4.8) shape the short-time spectrum of $\hat{s}[n]$ to match that of $s[n]$ in Figure 4.4.

measured by the analysis and the energy at the output of the $r$-th synthesis filter:

$$\begin{aligned} E_r(i) &= c_r(i)^2 \left( \frac{1}{K} \sum_{k=0}^{K-1} |H_r[k]|^2 \mathrm{E}\left\{ |\Psi[k]|^2 \right\} \right) && (4.9) \\ &= c_r(i)^2 \left( \frac{1}{K} \sum_{k=0}^{K-1} |H_r[k]|^2 \mathrm{E}\left\{ \sum_{n=0}^{N-1} \sum_{m=0}^{N-1} \psi[n]\psi[m] e^{j2\pi k(n-m)/K} \right\} \right) && (4.10) \\ &= c_r(i)^2 \sigma^2 \left( \frac{1}{K} \sum_{k=0}^{K-1} |H_r[k]|^2 \sum_{n=0}^{N-1} \sum_{m=0}^{N-1} \delta[n-m] e^{j2\pi k(n-m)/K} \right) && (4.11) \\ &= c_r(i)^2 \sigma^2 N \frac{1}{K} \sum_{k=0}^{K-1} |H_r[k]|^2 && (4.12) \\ &= c_r(i)^2 \sigma^2 N \sum_m h_r[m]^2, && (4.13) \end{aligned}$$

which can be manipulated to give Equation (4.8). The second interpretation is based on equalizing the short-time variances of the subband signal $s_r[n]$ and its estimate $\hat{s}_r[n]$. A slightly biased estimate of the variance of $s_r[n]$ in the $i$-th frame is given by [187]:

$$\mathrm{var}(s_{r,i}[n]) = \frac{1}{N} \sum_{n=0}^{N-1} s_r[n+iL]^2 = \frac{E_r(i)}{N}. \qquad (4.14)$$

The variance of $\hat{s}_r[n]$ in the $i$-th frame can be derived by considering the effect of a linear filter on the autocorrelation of a stochastic process:

$$\mathrm{E}\left\{ \hat{s}_r[n]\hat{s}_r[n+t] \right\} = \mathrm{E}\left\{ \sum_m c_r(i) h_r[m] \psi[n-m] \sum_l c_r(i) h_r[l] \psi[n+t-l] \right\} \qquad (4.15)$$

$$= \quad c_r(i)^2 \sum_m \sum_l h_r[m] h_r[l] \mathrm{E}\{\psi[n-m]\psi[n+t-l]\} \tag{4.16}$$

$$= \quad \sigma^2 c_r(i)^2 \sum_m \sum_l h_r[m] h_r[l] \delta[l-m-t] \tag{4.17}$$

$$= \quad \sigma^2 c_r(i)^2 \sum_m h_r[m] h_r[m+t]. \tag{4.18}$$

Evaluating at $t = 0$ yields the variance as

$$\mathrm{var}(\hat{s}_{r,i}[n]) \quad = \quad \sigma^2 c_r(i)^2 \sum_m h_r[m]^2. \tag{4.19}$$

Combining the expressions in (4.14) and (4.19) again yields the gain formula of Equation (4.8). This second perspective shows that this formulation does not involve strict process matching in the autocorrelation sense; rather, a loose matching is achieved in the sense that the local autocorrelations of the processes $s_r[n]$ and $\hat{s}_r[n]$ are equalized in the first order. In this light, the filter bank analysis-synthesis can be interpreted as a first-order subband linear predictive coding system. Higher order LPC methods, while designed to model locally stationary random processes, are not particularly useful for this modeling scenario since the parameterization is not tightly coupled to perceptual factors [100].

The formulation above can be rephrased in terms of the power spectral densities of the original and reconstructed processes. This provides a more intuitive explanation of the filter bank residual model than the variance matching framework, and relates the two interpretations given in the preceding paragraph. Using the Parseval relation

$$\sum_m h_r[m]^2 \quad = \quad \frac{1}{2\pi} \int_0^{2\pi} \left| H_r\left(e^{j\omega}\right) \right|^2 d\omega, \tag{4.20}$$

the subband gain from Equation (4.8) can be rewritten as

$$c_r(i)^2 \quad = \quad \frac{2\pi}{\sigma^2} \left[ \frac{E_r(i)}{N} \right] \frac{1}{\int_0^{2\pi} \left| H_r\left(e^{j\omega}\right) \right|^2 d\omega}. \tag{4.21}$$

The term $E_r(i)/N$ is a variance estimate as established in Equation (4.14); then, since the variance of a random process is the average value of its power spectral density (PSD), the gain can be further rewritten as

$$c_r(i)^2 \quad = \quad \frac{1}{\sigma^2} \frac{\int_0^{2\pi} S_{r,i}\left(e^{j\omega}\right) d\omega}{\int_0^{2\pi} \left| H_r\left(e^{j\omega}\right) \right|^2 d\omega}, \tag{4.22}$$

where $S_{r,i}\left(e^{j\omega}\right)$ is the PSD of the $r$-th subband signal in the analysis filter bank in the $i$-th frame. The numerator in the above expression can be written in terms of the PSD of the original signal $s[n]$:

$$c_r(i)^2 \quad = \quad \frac{1}{\sigma^2} \frac{\int_0^{2\pi} S_i\left(e^{j\omega}\right) \left| H_r\left(e^{j\omega}\right) \right|^2 d\omega}{\int_0^{2\pi} \left| H_r\left(e^{j\omega}\right) \right|^2 d\omega}. \tag{4.23}$$

This expression indicates that the gain for the $r$-th band is based on the average value of the input PSD over the $r$-th band; the $\sigma^2$ is necessary to normalize the variance of the white noise source $\psi[n]$ in the synthesis filter bank. Using the above equation, the PSD of the original and reconstructed signals can be related; note that the PSD of the synthesized process $\hat{s}_r[n]$ in the $i$-th frame is given simply by

$$\hat{S}_{r,i}\left(e^{j\omega}\right) \;=\; \sigma^2 c_r(i)^2 \left|H_r\left(e^{j\omega}\right)\right|^2. \tag{4.24}$$

Substituting for $c_r(i)$ yields

$$\hat{S}_{r,i}\left(e^{j\omega}\right) \;=\; \frac{\left|H_r\left(e^{j\omega}\right)\right|^2 \int_0^{2\pi} S_{r,i}\left(e^{j\omega}\right) d\omega}{\int_0^{2\pi} \left|H_r\left(e^{j\omega}\right)\right|^2 d\omega} \tag{4.25}$$

$$\;=\; \frac{\left|H_r\left(e^{j\omega}\right)\right|^2 \int_0^{2\pi} S_i\left(e^{j\omega}\right) \left|H_r\left(e^{j\omega}\right)\right|^2 d\omega}{\int_0^{2\pi} \left|H_r\left(e^{j\omega}\right)\right|^2 d\omega}. \tag{4.26}$$

This derivation shows that the ERB parameterization leads to a reconstruction whose subband power spectra correspond to averages of the input power spectra over the various bands of the filter bank. The formal relationship between the PSD of the full reconstructed signal and the original signal is more complicated, however, since cross terms are introduced in the output PSD because the subband signals are not independent. The constraints required to achieve such independence substantially restrict the filter bank design and are thus not incorporated; also, since the perceptual model is based on subbands, considerations regarding the PSD of the fully reconstructed output are not called for.

The result of Equation (4.8) clearly holds for the case $L = N$, where the gain is simply updated for each new synthesis frame. Abrupt gain changes at frame boundaries may cause discontinuities in the output; an alternative approach is to use $L = N/2$ and carry out an overlap-add process to construct the output. Then, the above gain calculation can also be applied, provided that the window overlap-adds to one for a stride of $N/2$ and that the energy in a given band does not change drastically from frame to frame.

This filter bank approach constitutes an effective framework for modeling the noiselike residual of the sinusoidal model in that it provides a small set of parameters that describe the general time-frequency behavior of the stochastic component. For example, the model is effective at the sample rate $f_s = 44.1kHz$ for $R = 12$ bands with a frame size of $N = 256$ and a stride of $L = 128$; in this case, the residual signal is essentially downsampled by a factor of 10 into a transparent parametric representation. The original and the synthesized signals have the same general time-frequency behavior, and because the ear is mostly insensitive to the fine details of a noiselike signal, this analysis-synthesis of the stochastic component is basically perceptually lossless. Greater compaction can be readily achieved by using larger frames and longer strides; the case above was cited in particular since it fits directly into the specific structure of the IFFT synthesizer discussed in Section 2.5. Also, to control the amount of model data, the number of bands can be increased or decreased simply by scaling the bandwidth of each ERB by a common factor. Finally, note that the length of the analysis frames and strides can be time-varied to estimate the residual parameters for a dynamically segmented sinusoidal model, *i.e.* a multiresolution model. Another way to generate model parameters at arbitrary times is to interpolate between data points taken at regularly spaced times; such interpolation assumes a certain smoothness in the evolution of the data.

### 4.2.3 Requirements for Residual Coding

The filter bank model of the sinusoidal analysis-synthesis residual meets three basic requirements for residual coding that have been established in the preceding discussions, namely compaction, perceptual relevance, and transparency. Compaction is especially desirable since the residual is secondary in importance to the primary reconstruction; perceptual relevance allows meaningful modifications to be carried out. Perceptual losslessness is of course useful for any audio signal model; in residual modeling, there is some leeway due to masking effects that occur upon combination with the primary reconstruction.

In addition to the criteria discussed above, another useful feature of a residual model is the ability to economically recombine the residual parameters with the parameters of the primary signal model prior to reconstruction. In the IFFT sinusoidal synthesizer, some computation is saved by using the ERB model to derive a spectral representation of the residual that can be combined with the sinusoidal spectrum before the IFFT. This FFT-based implementation is discussed further in the next section; a time-domain implementation of the filter bank model is also presented.

## 4.3 Residual Analysis-Synthesis

The filter bank model of broadband noise perception can be implemented in the time domain as formulated in the previous section. For frequency-domain sinusoidal synthesis, the model can be rephrased in terms of the FFT to allow a merged synthesis of the partials and the residual component. Details of both approaches are given below.

### 4.3.1 Filter Bank Implementation

The filter bank for the residual model is subject to looser design constraints than critically sampled filter banks. In this section, these constraints are discussed and a simple design approach is given; these formulations were originally presented in [178].

### Perfect reconstruction constraints

Perfect reconstruction filter banks were discussed at length in Section 2.2.1; recall that in the subsampled case, time-domain and/or frequency-domain aliasing introduced by the analysis is cancelled in the synthesis filtering process. Then, the requirement of a distortionless input-output transfer function along with this aliasing cancellation provides a set of design constraints for the filter bank. Due to the various advantages of subband processing, such filter bank approaches have been widely dealt with in the literature, but primarily for the case of uniform or octave-band filter banks [2, 20]. Some results on nonuniform critically sampled and oversampled perfect reconstruction filter banks have also been presented [188, 189, 190, 191, 192].

The design of a nonuniform filter bank for the noise perception model proposed in Section 4.2 differs from the perfect reconstruction problem discussed above. Summarizing the model, the ERB analysis filter bank provides a set of subband signals from which short-time gains are derived; in the synthesis, these gains are applied to the subbands of an ERB filter bank driven by white noise. In short, this scenario does not involve a typical critically sampled analysis-synthesis filter bank. In this framework, then, the filter bank design is subject to different constraints than those of a critically sampled system. A sensible perfect reconstruction constraint for the ERB filter bank is that the sum of the subband signals should equal the original signal; then, no distortion is introduced in deriving the subband ERB signals. For an $R$-band filter bank, this constraint corresponds simply to:

$$\sum_{r=1}^{R} s_r[n] \;=\; s[n] \quad\Longleftrightarrow\quad \sum_{r=1}^{R} h_r[n] \;=\; \delta[n]. \tag{4.27}$$

Scaling and delay are of course allowed since such effects can be readily compensated for in this application. Given the subband perfect reconstruction constraint, the only other issue is that arbitrary passband edges should be allowed for the filters at the design stage; such design flexibility enables a wider range of experiments, for instance with variable band allocation, than in a rigid approach. The filter bank design is discussed below.

### Filter bank design

Given a set of arbitrary frequency band edges spanning from 0 to the Nyquist frequency $f_s/2$, where the set will be denoted by

$$f_{\text{edges}} \;=\; \{f_0 \;\; f_1 \; \ldots \; f_r \; \ldots \; f_{R-1} \;\; f_R\} \tag{4.28}$$

with $f_0 = 0$ and $f_R = f_s/2$, which corresponds in radian frequency to

$$\omega_{\text{edges}} \;=\; \{\Phi_0 \;\; \Phi_1 \; \ldots \; \Phi_r \; \ldots \; \Phi_{R-1} \;\; \Phi_R\} \;=\; \frac{2\pi}{f_s} f_{\text{edges}} \tag{4.29}$$

with $\Phi_0 = 0$ and $\Phi_R = \pi$, consider ideal bandpass filters of the form

$$b_r[n] \;=\; \frac{\Delta_r}{\pi}\cos(\omega_r n)\left(\frac{\sin(\Delta_r n/2)}{\Delta_r n/2}\right), \tag{4.30}$$

where

$$\Delta_r \;=\; \Phi_r - \Phi_{r-1} \tag{4.31}$$

is the bandwidth of the $r$-th filter and

$$\omega_r \;=\; \frac{\Phi_r + \Phi_{r-1}}{2} \tag{4.32}$$

is the center frequency of the positive frequency passband of the $r$-th filter; because the filters are real, each has a negative frequency passband as well. Since the $R$ bands are nonoverlapping and span the entire spectrum by definition, the frequency responses $B_r(e^{j\omega})$ of the corresponding $R$ ideal bandpass filters simply add up to one:

$$\sum_{r=1}^{R} B_r\left(e^{j\omega}\right) \;=\; 1 \quad\Longleftrightarrow\quad \sum_{r=1}^{R} b_r[n] \;=\; \delta[n], \tag{4.33}$$

which shows that this ideal filter bank satisfies the subband perfect reconstruction constraint of Equation (4.27).

The ideal filter bank $\{B_r\left(e^{j\omega}\right)\}$ consists of two-sided IIR filters that are not realizable. However, a realizable FIR filter bank that satisfies the subband perfect reconstruction constraint can be derived from the ideal filter bank by using the *window method* of FIR filter design; this method suggests that a realizable FIR approximation of an ideal filter can be obtained by time-windowing the ideal filter's impulse response [193]. The frequency response of the approximate filter is given by a convolution of the ideal filter response and the transform of the window, which results in a smearing of the ideal response:

$$h_{\text{approx}}[n] \;=\; f[n]\,h_{\text{ideal}}[n] \;\Longleftrightarrow\; H_{\text{approx}}\left(e^{j\omega}\right) \;=\; F\left(e^{j\omega}\right) * H_{\text{ideal}}\left(e^{j\omega}\right). \tag{4.34}$$

This window-based approximation process is depicted in Figure 4.6; the approximate filter has *transition regions* in the frequency domain where the ideal filter has sharp cutoffs; also, ripples appear in the frequency response of the approximate filter.

In designing single filters, the window method leads to approximate realizations. In the filter bank case, however, it is possible to satisfy the subband perfect reconstruction condition *exactly* with realizable filters based on the window method. Introducing the window $f[n]$ on both sides of the right-hand expression in Equation (4.33) yields

$$f[n]\sum_{r=1}^{R} b_r[n] \;=\; \delta[n]f[n] \tag{4.35}$$

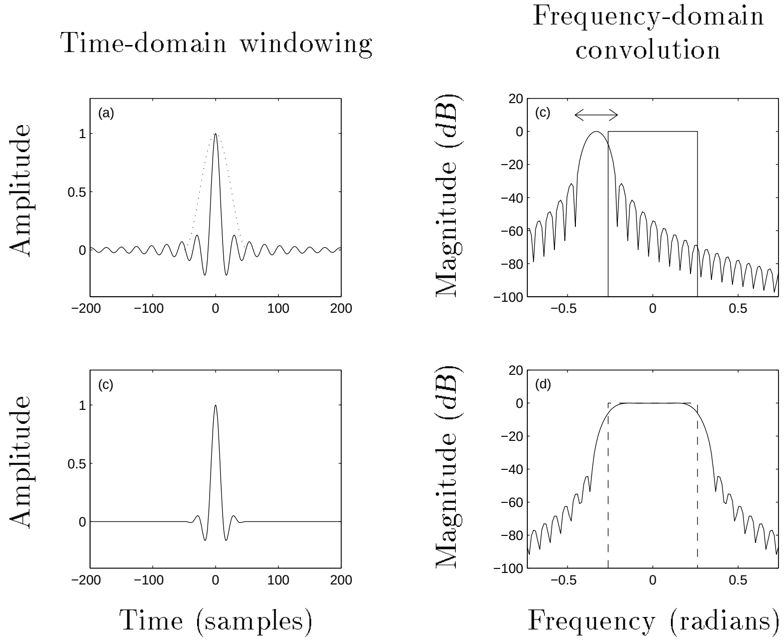

FIGURE 4.6: Window method for filter design. Time-domain multiplication corresponds to frequency-domain convolution, so windowing the sinc function as in (a) corresponds to the convolution shown in (b); the resulting FIR filter shown in (c) has the nonideal frequency response shown in (d), where the ideal filter (dashed) is included for comparison.

$$\Longrightarrow \sum_{r=1}^{R} f[n] b_r[n] \;=\; \sum_{r=1}^{R} h_r[n] \;=\; \delta[n] f[0], \tag{4.36}$$

where $h_r[n] = f[n]b_r[n]$. This verifies that the window-based filter bank also satisfy the constraint, provided that $f[0]$ is nonzero. In the frequency domain, application of the window corresponds to a convolution:

$$F\left(e^{j\omega}\right) * \sum_{r=1}^{R} B_r\left(e^{j\omega}\right) \;=\; F\left(e^{j\omega}\right) * 1 \tag{4.37}$$

$$\Longrightarrow \sum_{r=1}^{R} H_r\left(e^{j\omega}\right) \;=\; \frac{1}{2\pi}\int_0^{2\pi} F\left(e^{j\omega}\right) d\omega \;=\; f[0], \tag{4.38}$$

where $H_r(e^{j\omega}) \;=\; F(e^{j\omega}) * B_r(e^{j\omega})$. The convolution of $F(e^{j\omega})$ with the unity response of the ideal filter bank sum is simply equivalent to a full-band integration; the result of the integration is the constant $f[0]$. The nonideal filters $H_r(e^{j\omega})$ thus also satisfy the perfect reconstruction constraint of Equation (4.27); in the frequency-domain sum, the transition regions and ripples of a given filter are counteracted by contributions from the other filters. It should be noted here that this method does not readily apply to the design of subsampled perfect reconstruction filter banks.

The derivation in Equations (4.35) through (4.38) shows that the only restriction on the window $f[n]$ is that it be nonzero at $n = 0$; it can thus be used to vary the response of the filters without affecting the perfect reconstruction property. One useful choice for $f[n]$ is the raised cosine pulse, common in digital communication applications [124], which enables the filter responses to be controlled by way of the *excess bandwidth* parameter $\alpha$. The raised cosine is defined as

$$f[n] \;=\; \begin{cases} \dfrac{\cos\left(\frac{\Delta_1\alpha_1 n}{2}\right)}{1-\left(\frac{\Delta_1\alpha_1 n}{\pi}\right)^2} & -M \leq n \leq M \\ 0 & \text{otherwise} \end{cases} \tag{4.39}$$

such that the length of filters designed with this window is $2M+1$. Also, since the same window is applied to all the filters, the excess bandwidth parameter for the $r$-th filter is given by $\alpha_r\Delta_r = \alpha_1\Delta_1$. In choosing the excess bandwidth, there is thus only one degree of freedom, which implies that the overlap between adjacent filters will behave similarly across the entire spectrum. Filter bank responses based on this design are shown in Figure 4.7 for varying $M$ and $\alpha_1$, the excess bandwidth of the first filter.

The figure indicates the flexibility of the design: the band edges are arbitrary, the filter length is arbitrary but the same for each band, and the filter ripple and transition behavior are readily controllable. Beyond the standard time-frequency resolution tradeoffs in filter design, the flexibility of the filter response is limited only in that the formulation

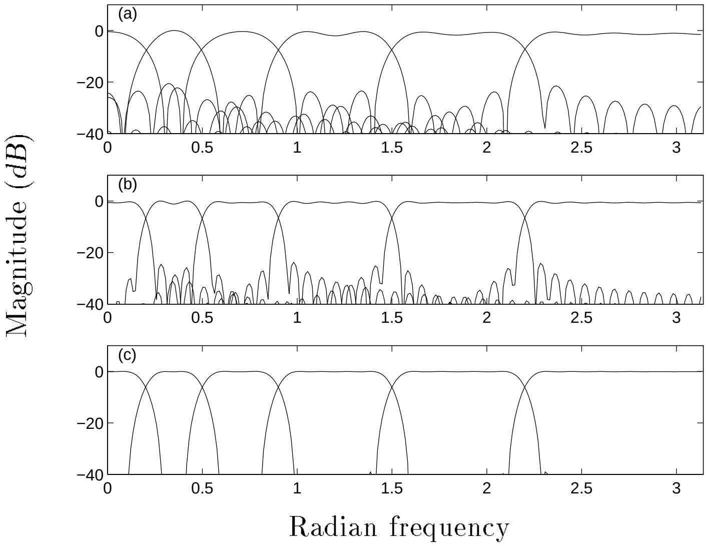

FIGURE 4.7: Frequency responses for a 6-band filter bank with (a) $M = 20$ and $\alpha_1 = 0.5$, (b) $M = 40$ and $\alpha_1 = 0.5$, and (c) $M = 40$ and $\alpha_1 = 0.95$.

requires that the same window function $f[n]$ be used for each filter in the filter bank. The choice of window essentially limits the frequency resolution of the narrowest band in the filter bank. For wide bands, the sinc impulse response is characteristically narrow, meaning that a long, smooth window will not affect the response drastically; for narrow bands, on the other hand, the time-domain sinc response is spread out. To maintain the frequency resolution of the narrowest band, then, it is necessary that the window be chosen long enough to cover the majority of the energy of the corresponding sinc function.

This design approach has proven useful for the ERB-based stochastic signal model; the ease and flexibility of the design allow for a wide variety of experiments involving reallocating the frequency bands and trading off the time-frequency resolution of the ERB parameterization.

### 4.3.2 FFT-Based Implementation

For the frequency-domain synthesizer discussed in Section 2.5, it is computationally advantageous to derive a representation of the residual that can be combined with the spectrum of the partials before the inverse Fourier transform is carried out. It is thus useful to devise an FFT-based algorithm for modeling the residual; analysis, synthesis, and normalization issues are discussed below. These results were presented in [110].

### Residual analysis

Analysis for the ERB residual model can be carried out using the FFT. As in the sinusoidal analysis, this uses a sliding window $w[n-iL]$ of length $N$ to extract frames of the residual $s[n]$ at times spaced by the analysis hop size $L$. The frame signal $w[n-iL]s[n]$ is then transformed into the spectrum $S(k,i)$ by a DFT of size $K$, where $K \geq N$. Note that the values of $N$, $K$, and $L$ need not correspond to those used in the sinusoidal analysis.

After the DFT, the spectrum is simply divided into bands according to the ERB model; without degradation of the model, the bandwidths of the ERBs can be scaled by a common factor to cover the spectrum with fewer bands and thereby achieve data reduction. After the band allocation is established, the energy in each of the bands is computed from the DFT magnitudes; the negative frequency components are not included since the spectrum is conjugate symmetric:

$$\tilde{E}_r(i) \;=\; \frac{1}{K}\sum_{k\in\beta_r} |S(k,i)|^2, \qquad (4.40)$$

where $\beta_r$ denotes the bins that fall in the $r$-th ERB; this shorthand will be used throughout the chapter. In this FFT-based analysis, these energies serve as the residual parameters for the $i$-th frame; changes in the characteristics of the residual are reflected in frame-to-frame variations of the ERB energies. Note that the energies $\tilde{E}_r(i)$ are not entirely the same as the $E_r(i)$ formulated in the filter bank analysis. However, both energy measures $E_r(i)$ and $\tilde{E}_r(i)$ are conceptually suitable for the psychoacoustic model, namely that the perceptual qualities of broadband noise are determined by the total energy in each band, and not by the specific distribution of energy within the bands. The distinction between $E_r(i)$ and $\tilde{E}_r(i)$ is discussed further later. Also note that the phase of the DFT $X[k]$ is irrelevant to the ERB energy calculation, which is justified since the auditory system is primarily sensitive to the magnitude of the short-time spectrum. This insensitivity to phase is especially applicable to the case of broadband noise, where the phase is itself a noiselike process; in such cases, the percept is basically independent of the phase distribution.

### Residual synthesis

The modeled residual can be synthesized with the IFFT as follows. First, the ERB energies are converted into a piecewise constant spectrum wherein the magnitude of each constant piece is determined by the corresponding ERB analysis parameter; these magnitudes correspond to the gains of the time-domain filter bank model. An example of this is given in Figure 4.8, which shows the magnitude spectrum of an analysis frame and the corresponding piecewise constant spectral estimate for synthesis based on twelve ERBs. Synthesis using piecewise linear spectral estimates, sloped within each ERB to fit the analysis spectrum, gives a reconstruction of the same perceptual quality as the

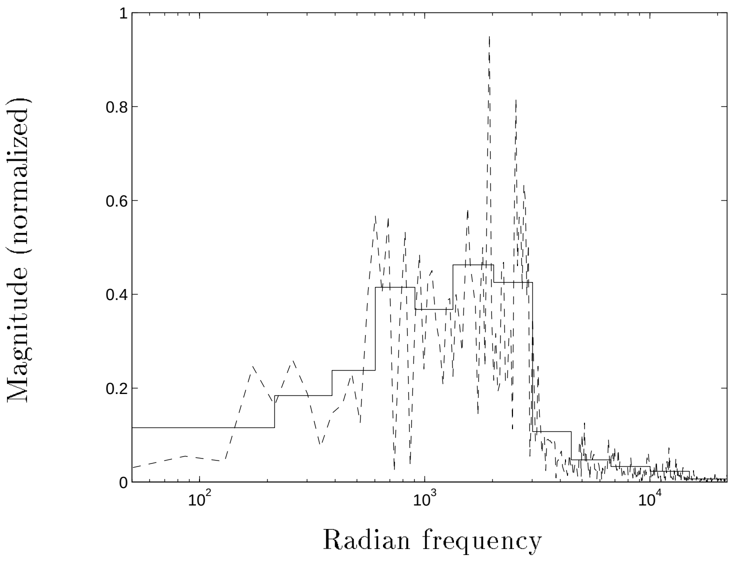

FIGURE 4.8: Piecewise constant ERB estimate (solid) of the residual magnitude spectrum (dotted) for a frame of a breathy saxophone note.

piecewise constant approach, which verifies the assumption that the ear is insensitive to the specific spectral distribution within each ERB.

For the sake of input-output equalization, it is important to preserve the ERB energies in the analysis-synthesis pathway; this is demonstrated by the following equations, where $S(k,i)$ denotes the analysis DFT for the $i$-th frame, $\hat{S}(\kappa,i)$ denotes the piecewise constant spectral estimate derived in the synthesis, $\xi_r$ is the number of bins in the $r$-th ERB at the synthesis stage, and $M$ is the size of the synthesis IFFT. Note that the analysis transform and the synthesis transform do not have to be the same size. Accordingly, the bins $\beta_r'$ in the $r$-th synthesis band are not necessarily the same as the bins $\beta_r$ in the $r$-th analysis band; also, distinct bin indices $k$ and $\kappa$ are used in the following formula:

$$\tilde{E}_r(i) \;=\; \frac{1}{M}\sum_{\kappa\in\beta_r'} |\hat{S}(\kappa,i)|^2 \;=\; \frac{1}{K}\sum_{k\in\beta_r} |S(k,i)|^2. \tag{4.41}$$

Every bin in a given synthesis band takes on the same value, so for any $\kappa \in \beta_r'$, the above equation can be rewritten as:

$$\tilde{E}_r(i) \;=\; \frac{\xi_r}{M}|\hat{S}(\kappa,i)|^2 \;\Longrightarrow\; |\hat{S}(\kappa,i)| \;=\; \sqrt{\frac{M}{\xi_r}\tilde{E}_r(i)}. \tag{4.42}$$

Energy preservation will be considered further in the upcoming section on normalization; specifically, normalization issues relating to the overlap-add synthesis process are dealt with there.

After the magnitude spectrum is constructed, a uniform random phase is applied on a bin-by-bin basis. Frame-to-frame phase correlations can be introduced to control the texture of the synthesized residual; for instance, varying the smoothness of the residual may be musically desirable. After the phase is incorporated, the spectrum of the residual model and the partial spectrum are summed (in rectangular coordinates) and transformed into a time-domain signal by the IFFT and OLA. This approach has proven perceptually viable for broadband residuals such as saxophone and flute breath noise.

### Comparison of FFT and filter bank analysis-synthesis methods

While apparently founded on the same basic psychoacoustic principle, the FFT-based model of the residual discussed in this section and the filter bank formulation of Section 4.2.2 provide different ERB energies for the model. Perceptually, the two methods yield similar results; given the allegation that the residual model is indeed different, it is of interest to compare the approaches mathematically so as to reveal the underlying issues.

The difference between the two methods can be understood in the framework of the STFT. Some restrictions must be imposed to compare the methods; these will be introduced as the framework is developed. In the FFT method, the analysis with the sliding window $w[n]$ can be immediately interpreted as a modulated STFT filter bank of the form shown in Figure 2.3, with analysis filters given by $w[-n]e^{j\omega_k n}$. Note that the difference between the ERB parameters depends on the analysis, so the synthesis filter bank will not enter the discussion here. From Section 2.2.1, the STFT of $s[n]$ with subsampling by $L$ is given by

$$S(k,i) \;=\; \sum_{n=0}^{N-1} w[n]s[n+iL]e^{-j\omega_k n}, \tag{4.43}$$

and the ERB parameters in the FFT method, as described earlier, are given by

$$\tilde{E}_r(i) \;=\; \frac{1}{K}\sum_{k\in\beta_r} |S(k,i)|^2 . \tag{4.44}$$

Then, summing the band energies across the spectrum yields the signal energy of Parseval's theorem:

$$\sum_{r=1}^{R} \tilde{E}_r(i) \;=\; \frac{1}{K}\sum_{r=1}^{R}\sum_{k\in\beta_r} |S(k,i)|^2 \;=\; \frac{1}{K}\sum_{k=0}^{K-1} |S(k,i)|^2 \;=\; \sum_{n=0}^{N-1} |w[n]s[n+iL]|^2 . \tag{4.45}$$

As will be seen, a similar summation does not generally apply in the filter bank case; the sum of the subband energies in a filter bank is not proportional to the energy of the original signal unless the filter bank corresponds to a tight frame [2].

In considering the filter bank approach, various restrictions must be imposed to allow for a meaningful comparison with the FFT method. First, the filters are restricted to be of the form

$$h_r[n] \;=\; f[n]b_r[n]e^{j\omega_r n}, \tag{4.46}$$

where $f[n]$ is a window function, $b_r[n]$ is an ideal filter, and $\omega_r = 2\pi k_r/K$, a bin frequency of a $K$-point FFT. Unlike earlier, these filters are defined to be complex; this allows for straightforward comparisons to the complex STFT filter bank. For real filters, a scale factor of two is simply necessary in some of the calculations to account for the negative frequency components.

With the above restriction on $\omega_r$ in mind, $b_r[n]$ is constrained to be of the form

$$b_r[n] \;=\; \sum_{k=-\epsilon_r}^{\epsilon_r} b[n]e^{j2\pi kn/K}, \tag{4.47}$$

where

$$b[n] \;=\; \frac{\sin(\pi n/K)}{\pi n}, \tag{4.48}$$

which corresponds in the frequency domain to an ideal filter of bandwidth $2\pi/K$, which is the width of one bin in a $K$-point FFT. The sum of modulated sinc functions in Equation (4.47) is then just an ideal filter of bandwidth $2\pi(2\epsilon_r+1)/K$. The last required restriction is that the window function $w[n]$ and the filter bank filters should be related by

$$w[n] \;=\; f[-n]b[-n]. \tag{4.49}$$

In other words, the FFT analysis window $w[n]$ is a windowed and time-reversed version of the impulse response of a narrowband sinc function. Given these restrictions, it is clear that any filter in the nonuniform filter bank corresponds to a sum of adjacent STFT filters:

$$\begin{aligned} h_r[n] \;&=\; f[n]b_r[n]e^{j\omega_r n} && (4.50)\\ &=\; f[n]e^{j\omega_r n}\sum_{k=-\epsilon_r}^{\epsilon_r} b[n]e^{j2\pi k/K} && (4.51)\\ &=\; \sum_{k=k_r-\epsilon_r}^{k_r+\epsilon_r} f[n]b[n]e^{j2\pi k/K} && (4.52)\\ &=\; \sum_{k=k_r-\epsilon_r}^{k_r+\epsilon_r} w[-n]e^{j2\pi k/K}. && (4.53)\end{aligned}$$

In this framework, the $r$-th subband signal of the ERB filter bank corresponds simply to

$$s_r[n] \;=\; \sum_{k\in\beta_r} S[k,n], \tag{4.54}$$

where the STFT $S[k,n]$ is not subsampled. The ERB energies in the filter bank approach are thus given by

$$E_r(i) \;=\; \sum_{n=iL}^{iL+N-1}\left|\sum_{k\in\beta_r} S[k,n]\right|^2. \tag{4.55}$$

In this case, the sum across bands does not yield the same result as the FFT method. This disparity occurs because of the nonlinearity of the magnitude function; the magnitude is taken at different points in the two methods. In the FFT method, the magnitude is taken before the subband signals are summed; in the filter bank method, the magnitude is taken after the subbands are added together.

The FFT and filter bank methods are mathematically distinct as derived above. However, they exhibit some type of equivalence in that the perceptual merits of the models are similar. This equivalence, despite the formal difference, indicates that a certain crudeness or inexactness can be incorporated into residual models without causing adverse effects; this is especially true if the inexactness is well-intentioned based on heuristics or simple psychoacoustics.

### An aside on Parseval's theorem

The filter bank residual model relies on the equivalence of time-domain and transform-domain signal energies; this equivalence is referred to as Parseval's theorem or relation. Parseval's theorem holds for any orthogonal basis, and a similar expression can be derived for the case of tight frames [2]. In this section, issues related to frequency-domain signal energies are considered. It should be noted that the issues to be discussed are not intrinsically coupled to the application of residual modeling, but indeed apply to arbitrary signals.

The frequency-domain representations of interest here are the discrete-time Fourier transform and the discrete Fourier transform. Considering Parseval's relation for these two cases leads to an interesting result. For an arbitrary discrete-time signal $x[n]$ of length $N$, the signal energy can be expressed in terms of the DTFT or the DFT, which is simply the uniformly sampled DTFT as discussed in Section 2.5.1:

$$\sum_{n=0}^{N-1} |x[n]|^2 = \frac{1}{2\pi}\int_{-\pi}^{\pi} \left|X\left(e^{j\omega}\right)\right|^2 d\omega \qquad \text{DTFT} \tag{4.56}$$

$$= \frac{1}{K}\sum_{k=0}^{K-1} |X[k]|^2 \qquad \text{DFT with } K \geq N \tag{4.57}$$

$$= \frac{1}{K}\sum_{k=0}^{K-1} \left|X\left(e^{j2\pi k/K}\right)\right|^2. \tag{4.58}$$

The right-hand expressions in Equations (4.56) and (4.58) can be equated and manipulated into the form

$$\int_{-\pi}^{\pi} \left|X\left(e^{j\omega}\right)\right|^2 d\omega = \sum_{k=0}^{K-1} \left|X\left(e^{j2\pi k/K}\right)\right|^2 \left(\frac{2\pi}{K}\right). \tag{4.59}$$

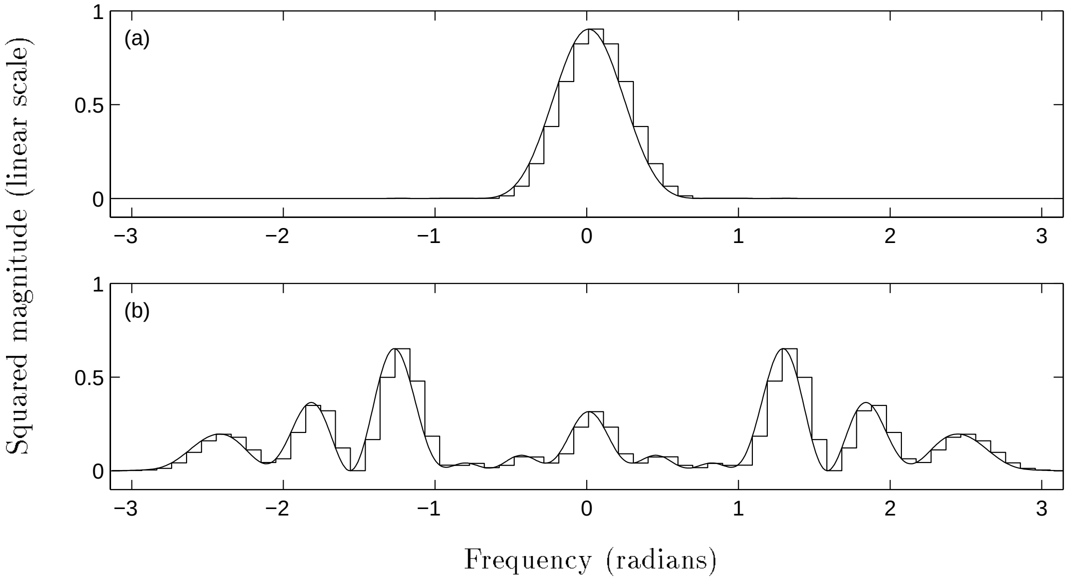

FIGURE 4.9: Examples of exact stepwise integration for two spectra. As shown in Equation (4.59), Parseval's theorem indicates that the stepwise approximation of the DTFT squared-magnitude based on the DFT is exact if the DFT is large enough that no time-domain aliasing is introduced.

The left side is simply the integral of the magnitude-squared of the DTFT. The right side can be interpreted as a piecewise approximation of the continuous integral; the width of a piece is $2\pi/K$ and the height of the piece spanning from frequency $2\pi k/K$ to $2\pi(k{+}1)/K$ is $|X[k]|^2$, the squared magnitude of the DTFT sample at $2\pi k/K$. This stepwise integration is illustrated in Figure 4.9. For the squared magnitude of the DTFT, there is no error in approximating the integral in this fashion as long as the DFT is large enough, *i.e.* there are enough samples of the DTFT. Essentially, this condition holds because the signal is time-limited; the notion is analogous to the familiar result that a bandlimited signal can be perfectly reconstructed from an appropriate set of samples [194]. This issue is mostly an aside from the discussion of residual modeling, so it will not be considered further.

### Normalization

To achieve perceptual losslessness in a deterministic-plus-stochastic or reconstruction-plus-residual model, it is necessary that the relative perceptual strengths of the two components be preserved by the system. The STFT peak picking described in Chapter 2 provides the proper amplitudes for equalized sinusoidal synthesis. In the residual model, the loudness equalization is based on preserving the short-time energy of the signal (in a stochastic sense); such energy preservation was the basis for deriving the short-time gains for the synthesis filter bank discussed in Section 4.2.2. In the frequency-domain synthesizer, the various operations mandate careful considerations of their effects on the short-time signal energy. Relative equalization of the subband energies is straightforward as derived earlier; the various windowing and overlap-add operations, however, introduce gain changes that must be compensated for.

The FFT-based residual analysis-synthesis is depicted in Figure 4.10. With the exception of the transparency requirement, the ERB energy parameters in this model immediately meet the criteria discussed in Section 4.2.3; namely, the ERB energies comprise a small set of perceptually meaningful parameters that can be readily combined with the partials before the IFFT. To meet the final requirement of perceptual losslessness, signal scaling must be explicitly accounted for; due to the multiple windowing steps and the possibility of different analysis and synthesis frame sizes and sampling rates, the synthesized residual may not have the same loudness as the original residual. In the following derivations, the subscripts $a$ and $s$ refer to the analysis and synthesis stages, respectively.

The proper scaling of the residual can be derived by considering the energy in the continuous-time signal. For an input segment of length $\tau_a$ corresponding to $N$ samples at the rate $f_a = 1/T_a$, the energy in the continuous-time signal is

$$E'_a \;=\; \int_{\tau_0}^{\tau_0+\tau_a} s(t)^2 dt \;\approx\; \sum_{n=0}^{N-1} s[\tau_0 + nT_a]^2 T_a, \tag{4.60}$$

where the $\approx$ refers to the approximation of the integral by the sum of the areas of rectangles of width $T_a$ and height $s[\tau_0 + nT_a]^2$. In discrete time, the energy of this analysis frame of length $N$ is

$$E_a \;=\; \sum_{n=0}^{N-1} s[n]^2 \;=\; \frac{E'_a}{T_a}. \tag{4.61}$$

The expected value of this energy, which will indeed be used as a measure of energy in the following, is simply given by

$$\mathrm{E}\{E_a\} \;=\; \sum_{n=0}^{N-1} \mathrm{E}\{s[n]^2\} \;=\; N\mathrm{E}\{s[n]^2\}. \tag{4.62}$$

This frame energy is now traced through the system; note that, as before, a frame index is dropped without loss of generality.

First, the output $w[n]s[n]$ of the analysis window has energy

$$E_w \;=\; \sum_{n=0}^{N-1} w[n]^2 s[n]^2. \tag{4.63}$$

Similarly to the derivation for the time-domain filter bank, the expected value of the energy is now used as a metric; replacing $s[n]^2$ by its expected value in Equations (4.61) and (4.63) gives

$$E_w \;=\; E\{s[n]^2\} \sum_{n=0}^{N-1} w[n]^2 \;=\; \frac{\mathrm{E}\{E_a\}}{N} \sum_{n=0}^{N-1} w[n]^2, \tag{4.64}$$

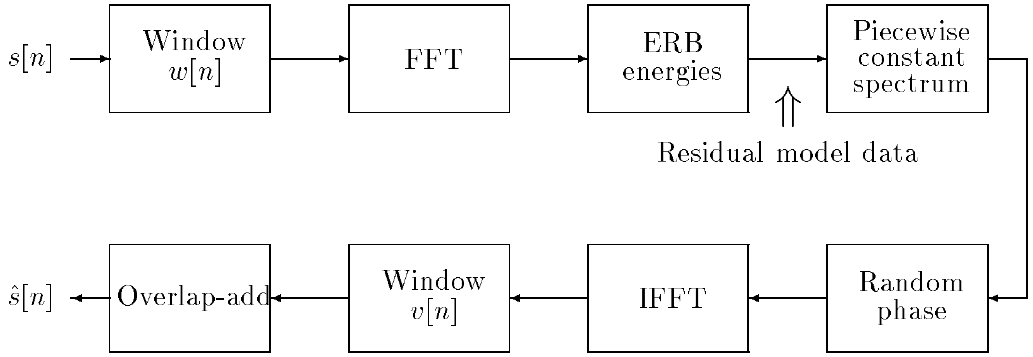

FIGURE 4.10: Block diagram of the FFT-based residual analysis-synthesis. The first three blocks constitute the analysis.

which indicates how the windowing process affects the signal energy. By Parseval's theorem, the $K$-point analysis FFT preserves this energy measure, as does the ERB energy estimation, by construction; the $M$-point IFFT likewise preserves the energy as long as the spectrum is constructed according to (4.42). Using a similar argument as for the analysis window, the effect of OLA with the length-$M$ window $v[n]$ can be shown to be

$$E_s = \frac{2E_w}{M} \sum_{i=0}^{M-1} v[n] \left( v[n] + v_1[n] + v_2[n] \right), \tag{4.65}$$

where $v_1[n]$ and $v_2[n]$ correspond to the second half of the window from the previous frame and the first half of the window from the subsequent frame, respectively:

$$v_1[n] = \begin{cases} v\left[n + \frac{M}{2}\right] & 0 \leq n < \frac{M}{2} \\ 0 & \frac{M}{2} \leq n < M \end{cases} \tag{4.66}$$

$$v_2[n] = \begin{cases} 0 & 0 \leq n < \frac{M}{2} \\ v\left[n - \frac{M}{2}\right] & \frac{M}{2} \leq n < M. \end{cases} \tag{4.67}$$

Note that a 50% overlap factor has been assumed in the derivation. As a check on the accuracy of this formulation, for a window $v[n]$ that overlap-adds to one, the post-windowing and OLA do not affect the energy. In this FFT-based synthesis system, the ERB spectrum is added to the partial spectrum before the IFFT; the effective OLA window $v[n]$ for the residual is then a triangular window divided by the motif window. This hybrid window, discussed in Section 2.5, does not overlap-add to one; for this reason, the OLA scale factor must be included.

The energy $E_s$ given by Equation (4.65) is the discrete-time energy for a synthesis frame of length $M$. The energy of the continuous time output signal $\hat{s}(t)$ is

$$E_s' = \int_{\tau_0}^{\tau_0+\tau_s} \hat{s}(t)^2 dt \tag{4.68}$$

$$\approx \sum_{n=0}^{M-1} \hat{s}[\tau_0 + nT_s]^2 T_s = E_s T_s, \tag{4.69}$$

where $T_s$ is the synthesis sampling period and $\tau_s$ is the duration of the $M$-sample output frame: $\tau_s = MT_s$. However, since the input energy corresponds to an input segment of duration $\tau_a$, what is required is an equalization of the energy for an output segment of that same duration $\tau_a$. Let $O$ denote the number of output samples (at rate $1/T_s$) in a segment of duration $\tau_a$; the energy in this segment is

$$E_s'' = \sum_{n=0}^{O-1} \hat{s}[n]^2 T_s = \frac{O}{M} E_s'. \tag{4.70}$$

Noting that

$$\frac{O}{M} = \frac{\tau_a}{\tau_s} \implies \frac{O}{M}\frac{T_s}{T_a} = \frac{N}{M} \tag{4.71}$$

the entire transformation of the continuous time energies can be expressed as

$$E_s'' = G_s G_a E_a', \tag{4.72}$$

where

$$G_a = \sum_{n=0}^{N-1} w[n]^2 \tag{4.73}$$

is the energy scaling incurred in the analysis, and

$$G_s = \frac{2}{M^2} \sum_{m=0}^{M-1} v[m] \left( v[m] + v_1[m] + v_2[m] \right) \tag{4.74}$$

is the effect of synthesis. In the analysis, then, the signal should be multiplied by the scale factor $1/\sqrt{G_a}$ before the ERB energies are calculated; at the synthesis stage, the output should be multiplied by $1/\sqrt{G_s}$ to equalize the energies. Listening tests have verified that the signal energy of Parseval's theorem is an accurate measure of the loudness of broadband noise, and that the outlined approach provides input-output equalization in the ERB analysis-synthesis.

## 4.4 Conclusion

In modeling complicated signals, it is often necessary to introduce a mixture of representations. This chapter described the specific framework of residual modeling, in which the signal is first reconstructed based on a primary model, and the difference between the original and the reconstruction is then modeled independently. For the multiresolution sinusoidal model, this residual is a colored noise process that can be parameterized in a perceptually accurate fashion in terms of the subband energies of the auditory filters; in audio applications, this process includes features that are perceptually important for realism, *e.g.* breath noise in a flute. This chapter discussed basic filter bank models of the auditory system as well as a simple approach for designing corresponding filter banks. Two implementations of the resulting residual model were developed and compared. It was shown that the parameterizations in the two implementations are somewhat different but further argued that the difference is practically moot; this contention is based on the experimental observation that crude heuristic models of noiselike signals can achieve transparency. It should be noted that the residual model discussed in this chapter is not signal-adaptive; rather, it is intended for use in conjunction with signal-adaptive sinusoidal models. Of course, the model could be made signal-adaptive by using a filter bank with adaptive band allocation, for instance, but such adaptation has not proven necessary for modeling typical residuals.

# Chapter 5

# Pitch-Synchronous Methods

In the general sinusoidal model, the frequencies of the partials are estimated without regard for the possibility of harmonic structure; at least, it is not necessary to make any assumptions about the presence of such behavior. In cases where harmonic structure is prevalent, *i.e.* in periodic and pseudo-periodic signals, this can be exploited to improve the signal model with respect to data reduction in that only the fundamental frequency need be recorded. In this chapter, a pitch-synchronous signal representation proposed in [195] is considered; similar representations have been applied in prototype waveform speech coders [56]. This pitch-dependent framework leads to simple sinusoidal models in which line tracking and peak detection are unnecessary because of the harmonic structure; furthermore, the representation leads to wavelet-based models that are more appropriate for pseudo-periodic signals than the lowpass-plus-details model of the standard discrete wavelet transform. By separately estimating the pitch or periodicity of a signal, improvements in both wavelet and sinusoidal models can be achieved. It should be noted that these approaches rely on robust pitch detection and thus apply only to signals whose periodic structure can be reliably estimated; in audio applications, then, appropriate signals consist of a single voice or a single instrument.

## 5.1 Pitch Estimation

Pitch estimation or *pitch detection* refers to the problem of finding the basic repetitive time-domain structure within a signal. This issue has been explored most extensively in the speech and audio processing communities [1, 53, 196, 197, 198]; the terminology is thus taken from these fields, but the methods apply to any pseudo-periodic signals. Pitch detection is reviewed in the section below; the section thereafter proposes a simple algorithm for refining pitch estimates for the purpose of carrying out pitch-synchronous signal segmentation.

### 5.1.1 Review of Pitch Detection Algorithms

Algorithms for pitch detection can be loosely grouped into time-domain and frequency-domain methods. In frequency-domain approaches, a short-time spectrum of the signal is analyzed for harmonic behavior, *i.e.* peaks in the spectrum at frequencies with a common factor; this factor corresponds to the fundamental frequency of the signal. In time-domain techniques, cross-correlations of nearby signal segments are computed at various lags; the lags that yield peaks in the cross-correlation correspond to the period of the signal. Both types of methods are fundamentally susceptible to errors: for instance, in the time domain, a two-period signal segment can be mistaken as a pitch period; in the frequency domain, dominance of either the odd or even harmonics, or a missing fundamental, can result in significant estimation errors. Various fixes have been proposed to account for these problems; for instance, based on the *a priori* knowledge that a typical musical signal does not have impulsive pitch discontinuities, a median filter can be applied to the pitch estimates to remove outliers and provide a more robust estimate [1, 53, 197].

For a more detailed discussion of pitch detection algorithms, the reader is referred to [1, 53, 196]. For the purposes of this chapter, it is assumed that a reliable pitch detection algorithm is available, and that the algorithm is capable of acknowledging, perhaps according to some heuristic threshold, when no pitch can be reasonably assessed to the signal. Using this assessment, the algorithm can segment the signal into regions classified as *pitched* or *unpitched.*

### 5.1.2 Phase-Locked Pitch Detection

A standard pitch detector provides an estimate of the local pitch of a signal, which is essentially a rough parametric description of the local behavior. A rough estimate of the local behavior is not entirely adequate, however, for the applications to be discussed here; as will be seen, it is important that the pitch estimates correspond to precise structures in the signal. To achieve this correspondence, pitch estimates from a standard algorithm can be "phase-locked" to the signal as proposed below. First, it is assumed that a robust pitch detector such as the one described in [197] is used to generate a moving estimate of the pitch period; the output of the pitch detector is specifically assumed to consist of pitch periods and their corresponding time indices. This pitch period function will be denoted by $P(t)$; since detectors generally estimate the pitch at some fixed interval $T$, the function $P(t)$ can be equivalently represented as $P(iT) = P(t)|_{t=iT}$. It is further assumed for the sake of notation that the pitch detection algorithm assigns a value of zero to $P(t)$ when no reasonable pitch can be assessed to the signal. Note that the onset of a signal cannot typically be assigned a pitch, so $P(t) = 0$, or likewise $P(iT) = 0$, will generally be the case in the onset regions; after the onset, if the signal becomes pseudo-periodic a pitch can be estimated. A similar observation holds for transitions, for instance note-to-note changes in music; a pitch cannot be assigned to the interstitial regions. Given these assumptions and observations, the phase-locking algorithm is straightforward; it is explained here as well as in the flowchart of Figure 5.1:

- For the first pitch detected after a region where $P(iT) = 0$, find the corresponding time point in the signal ($t_a$) and search for the first subsequent positive-slope zero crossing in the signal. Denote this by $t_0$. Since time is discretized and the zero crossing may not fall on a sample point, $t_0$ is chosen to correspond to the first positive signal value after the zero crossing.

- The time $t_0$ lies between two times $t_a$ and $t_b$ for which pitches have been estimated by the initial pitch detection algorithm; thus, an appropriate estimate of the pitch period at time $t_0$ can be found by interpolating:

$$P(t_0) \;=\; \frac{P(t_a)[t_b - t_0] + P(t_b)[t_0 - t_a]}{t_b - t_a}. \tag{5.1}$$

  $P(t_0)$ is the estimated length of the signal period starting at $t_0$.

- Find the positive-slope zero crossing closest to (not necessarily after) the time $t_0 + P(t_0)$. Denote this time by $t_1$. Again, the time is rounded to correspond to the positive value after the zero crossing.

- Interpolate to estimate $P(t_1)$, and then find $t_2$, which is the time of the closest positive-slope zero crossing to $t_1 + P(t_1)$.

- Repeat the above step for $t_2$, and so on, until a region where $P(iT) = 0$ is entered, at which point the algorithm should be restarted entirely.

- At stages in the interpolation when $P(t_a) \neq 0$ and $P(t_b) = 0$, the interpolated pitch is assigned a zero value to prevent incongruous pitch estimates.

- The time points $\{t_1, t_2, \ldots\}$ indicate pitch period boundaries that can be used to construct a track of phase-locked period estimates $\hat{P}(t_j) \;=\; t_{j+1} - t_j$. The starting times of the pitch periods follow positive-slope zero crossings by construction, so the first sample in any pitch period is positive and the last sample is negative.

This phase-locking algorithm yields a set of refined pitch period estimates that correspond to pseudo-periodic structures that are synchronized to positive-slope zero crossings of the signal; as will be seen, synchronization at zero crossings, while seemingly arbitrary, is of importance for deriving a useful pitch-synchronous signal representation. Furthermore, it has also been reported that zero crossings are of physical significance in speech signals in that they are linked to instances when the glottis is closed [198].

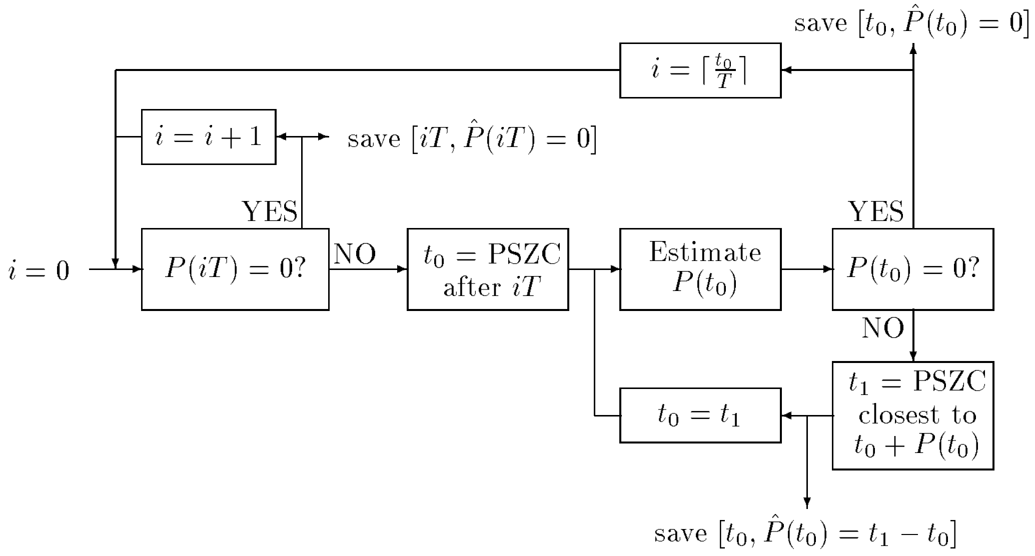
save $[t_0, \hat{P}(t_0) = 0]$
$i = \lceil \frac{t_0}{T} \rceil$
$i = i + 1$
save $[iT, \hat{P}(iT) = 0]$
YES
$i = 0$
$P(iT) = 0?$
NO
$t_0 =$ PSZC after $iT$
Estimate $P(t_0)$
YES
$P(t_0) = 0?$
NO
$t_1 =$ PSZC closest to $t_0 + P(t_0)$
$t_0 = t_1$
save $[t_0, \hat{P}(t_0) = t_1 - t_0]$

FIGURE 5.1: Flow chart for phase-locked pitch detection. The abbreviation PSZC refers to a positive-slope zero crossing in the signal. It is assumed that initial pitch period estimates, denoted by $P(iT)$, are derived by a standard pitch detection algorithm such as the one described in [197]. The itemized description in the text gives additional details related to the operations carried out by the various blocks.

Some wavelet-based algorithms for pitch estimation based on zero crossings have been discussed in the literature [199, 198]; the corrective phase-locking described above is adhered to in this treatment, however, since it is simple and allows for a quick synchronization of pitch period estimates to zero crossings in the signal.

## 5.2 Pitch-Synchronous Signal Representation

Using the time points from the simple phase-locked pitch detector presented above, the signal can be divided into pseudo-periodic segments, *i.e.* pitch periods that are synchronized to positive-slope zero crossings. This segmentation leads to a pitch-synchronous representation similar to the one proposed in [195]; this representation will prove useful for signal modeling.

### 5.2.1 Segmentation

In Section 4.1, mixed models of signals were discussed; this motivated considering the sinusoidal model in terms of a deterministic-plus-stochastic decomposition where the stochastic component accounted for signal features not well-represented by the sinusoidal model. The overall model mixture then consisted of slowly-varying sinusoids and broadband noise.

A representation similar to the deterministic-plus-stochastic decomposition of Chapter 2 has been widely applied in linear predictive coding (LPC) of speech, where the speech is coded using a time-varying source-filter model [1, 23]. The filter is adapted in time to match the speech spectrum, while the source is chosen based on a classification of the local speech signal as voiced or unvoiced. The characterization *voiced* refers to sounds, such as vowels, that exhibit a strong periodicity; the corresponding source for the LPC model is a periodic impulse train. The alternative classification *unvoiced* designates sounds such as sibilants and fricatives which do not exhibit periodic behavior and are heuristically more noiselike; the source for unvoiced sounds is typically white noise. Synthesis in the LPC framework is carried out by applying the appropriate source to the time-varying filter; when the input is a periodic impulse train, the output has the pseudo-periodic structure characteristic of voiced sounds, whereas when the input is white noise, the output is simply colored noise and does not exhibit periodicities.

In LPC, the voiced/unvoiced classification parameter indicates a segmentation of the signal into regions where different models are appropriate. A similar segmentation can be applied to arbitrary audio signals; because the terms "voiced" and "unvoiced" are inappropriate designations for musical signals, the terms "pitched" and "unpitched" will be used to classify the signal behavior. The phase-locked pitch detection algorithm described in the previous section is appropriate for deriving such a pitched/unpitched signal segmentation; regions where a pitch can be estimated are designated as pitched and regions where $P(t) = 0$ are classified as unpitched. This segmentation is markedly different from the deterministic-plus-stochastic decomposition described in the treatment of the sinusoidal model; as discussed in Section 4.1, in the sinusoidal model and in some LPC variations, the model mixtures are concurrent in time. For pitch-synchronous processing, however, it is necessary to neglect such concurrency and rigidly segment the signal into pitched and unpitched regions. As will be seen, this introduces some difficulties in the modeling of unpitched transient regions; a resolution of these difficulties is arrived at in Section 5.4.3.

In segmenting a dynamic signal such as a musical phrase, the transitions between regions of different pitch are classified as unpitched as described above. Pitch-synchronous processing algorithms are adjusted at these transitions to account for the pitch variations. In addition to variations across transitions, each local pitch region exhibits variations, for instance those that accompany vibrato; such variations are natural in musical signals and occur even when a vibrato is not immediately perceptible. For the algorithms to be discussed, it is necessary to segment the signal into pitch periods within each local pitch region. Because of signal characteristics such as vibrato, however, these pitch period segments do not each have the same duration. As will be seen, it is necessary to remove these local pitch variations prior to processing; the variations can be reinjected in the synthesis if necessary for realism. Removal of local pitch variations is described in the next section.

### 5.2.2 Resampling

In general digital audio applications, it is often desirable to change the sampling rate; this can be done straightforwardly by converting the signal to continuous time and then sampling at the desired rate, but that approach is both inefficient and not robust to noise degradations. It is thus of interest to effect a change in sampling rate in the digital domain. This process is referred to as sample rate conversion or *resampling*. For the applications in this chapter, resampling will be used to remove local pitch variations prior to carrying out pitch-synchronous processing; as stated above, the pitch variations can be reintroduced at the synthesis stage if perceptually necessary.

One method of resampling uses the familiar upsampling and downsampling operations. Changing the sampling rate of a sequence $x[n]$ from $f_s$ to $\frac{P}{Q}f_s$ is carried out by upsampling by $P$ and then downsampling by $Q$, with some appropriate intermediate filtering to prevent aliasing [193]. The resulting sequence is $\frac{P}{Q}$ times as long as $x[n]$. A detailed consideration of this type of approach can be found in [200].

In the pitch-synchronous methods of this chapter, resampling is carried out for each pitch period of the signal so as to remove slight pitch variations; this enables construction of the pitch-synchronous signal representation discussed in the next section, which will prove useful for signal coding and modification. The idea is simply to take the local pitch period segments and resample each one to some period $P$. In the following discussion, then, $P$ will serve to denote the target period; $Q_i$ will denote the original period of the $i$-th pitch period segment, which will be referred to as $x_i[n]$. Finally, $R$ denotes the number of pitch period segments in the local pitch region.

Resampling using the filter-based approach described above tends to introduce edge effects. This is problematic for the application of pitch period resampling since it tends to result in discontinuities at period boundaries in the signal reconstructions in the various pitch-synchronous models to be presented. An alternative method based on the discrete Fourier transform is more appropriate for this resampling application since it introduces fewer artifacts at signal boundaries.

Resampling using the DFT is carried out as follows [201]. For a pitch period $x_i[n]$ of length $Q_i$, a DFT of size $Q_i$ is computed, unless of course $Q_i$ is equal to the target period $P$. The spectrum is then truncated or extended to size $P$ as described in the following list, and an IDFT of size $P$ scaled by $\frac{P}{Q_i}$ yields the output sequence $x_i'[n]$ of length $P$. The resized spectrum is derived differently depending on the relative values of $P$ and $Q_i$:

- $P = Q_i$. No resampling is necessary. Since this is computationally advantageous, the target period $P$ for a local pitch region is chosen as the mode of the original periods $\{Q_i, i \in [1, R]\}$ so that this case occurs frequently.

- $P < Q_i$. The resampled output is to be shorter than the input, so the modified spectrum should have fewer bins than the original. This is carried out by discarding the $P - Q_i$ highest frequency bins, which is equivalent to eliminating the highest frequency harmonics from the signal.
- $P > Q_i$. The resampled output is to be longer than the input, so the modified spectrum should have more bins than the original. This is done by introducing $P - Q_i$ high-frequency harmonics having either zero amplitude or nonzero amplitudes derived by extrapolating the original spectrum.

Note that the Nyquist frequency bin, if present (when $P$ or $Q_i$ is odd), is always zeroed out. Also note that since the sampling rate is necessarily large in high-quality audio applications, the periods $P$ and $Q_i$ are both typically fairly large. Since local pitch variations are typically small with respect to the average local pitch, the spectral adjustments described above are relatively minor. The DFT computation, however, may be intensive, especially if $P$ or $Q_i$ is prime. The cost is not prohibitive, however, since the algorithms to be discussed are intended primarily for off-line use. Further treatment of resampling is not merited here; for the remainder of the chapter, it is assumed that the pitch variations can be reliably removed.

### 5.2.3 The Pitch-Synchronous Representation Matrix

Once the pitch variations in the $R$ pitch period segments have been removed via resampling, the signal can be reorganized into an $R \times P$ matrix

$$X = \begin{bmatrix} x'_1[n] \\ x'_2[n] \\ x'_3[n] \\ \vdots \\ x'_R[n] \end{bmatrix}, \qquad (5.2)$$

where $x'_i[n]$ is a version of the pitch period $x_i[n]$ that has been resampled to length $P$. The matrix will be referred to as the pitch-synchronous representation (PSR) of the signal. As described in the next section, this representation is useful for carrying out modifications; furthermore, structuring the signal in this fashion leads to the pitch-synchronous sinusoidal models and wavelet transforms discussed later.

There are several noteworthy issues regarding the PSR. For one, the matrix need not be constructed via resampling. Alternatively, the period lengths can be equalized by zero-padding all of the period signals to the maximum period length [195] or by viewing each period as an impulse response and carrying out an extension procedure such as in pitch-synchronous overlap-add methods [90]. These approaches, however, do not yield the same smoothness as resampling; they do not necessarily preserve the zero-crossing synchronization and discontinuities may result in the reconstruction.

A second issue concerns the unpitched regions. Each pitched region in a signal has a preceding unpitched region; this structure allows the approach to be readily generalized from the single note scenario to the case of musical phrases. Given this argument, the considerations herein are primarily limited to signals consisting of a single note. In the single-note case, the preceding attack is then the unpitched region in question. To allow for uniform processing of the signal, the attack is split into segments of length $P$ and included in the PSR; the beginning of a signal is zero-padded so that the length of the onset is a multiple of $P$. In later sections, perfect reconstruction of the attacks is considered in the frameworks of both pitch-synchronous Fourier and wavelet models. In either of the transforms, the signal is reconstructed after processing by concatenating the rows of the synthesis PSR, possibly resampled to the original pitch periods using pitch side information if necessary for realism.

An example of a PSR matrix is given in Figure 5.2 for a portion of a bassoon note. This bassoon signal and variations of a similar synthetic signal will be used throughout this chapter to illustrate the issues at hand. Note that the PSR is immediately meaningful for signals consisting either of single notes or several simultaneous notes that are harmonically related. For musical phrases or voice, it is necessary to generate a different PSR for each pitch region in the signal; the various PSR matrices have different dimensions depending on the local pitch and duration of that pitch. This chapter focuses on the single-pitch case without loss of generality; extensions of the algorithms are straightforward.

### 5.2.4 Granulation and Modification

The pitch-synchronous representation is a granulation of the signal that can be readily used to facilitate several types of modification: time-scaling, pitch-shifting, and pitch-synchronous filtering. First, time-scaling can be carried out by deleting or repeating pitch period grains for time-scale compression or expansion, respectively; this can be done either in a structured fashion or pseudo-randomly. In speech processing and granular synthesis applications, similar techniques are referred to as *deletion* and *repetition* [202, 203]. Note that the time-scaling by deletion/repetition is accomplished without pitch-shifting, and that it is inherently made possible by the zero-crossing synchronization of the PSR; without this imposed smoothness of the model, discontinuities would result in the modified signal.

Pitch-shifting based on the PSR is done simply by resampling the pitch periods; such pitch-shifting is not formant-corrected, however, but formant correction, which was

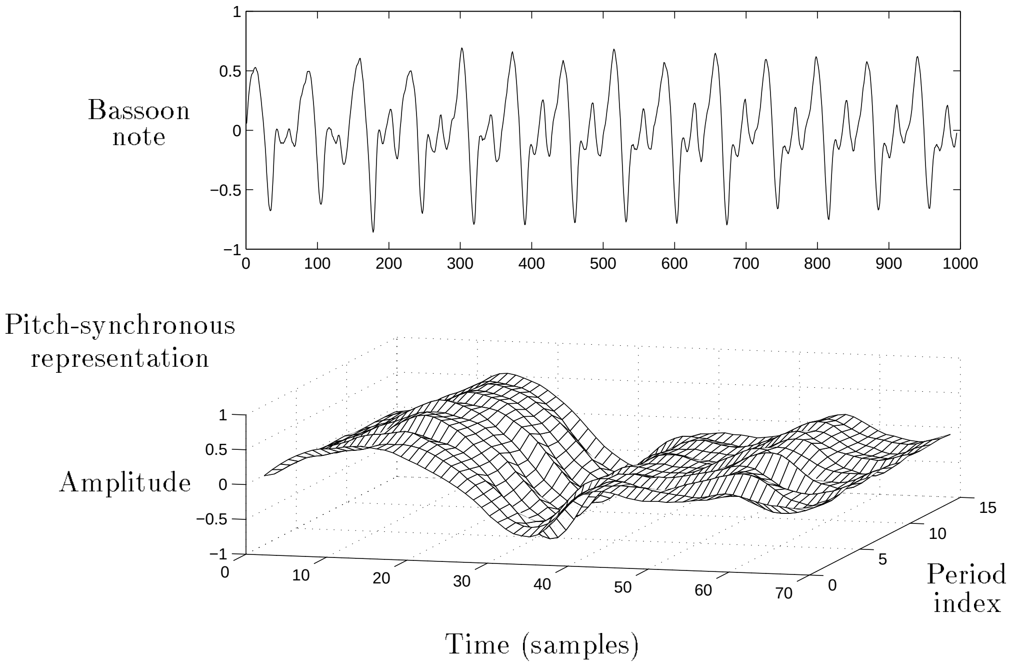

FIGURE 5.2: A portion of a bassoon note and its pitch-synchronous representation.

discussed in Section 2.7.2, can be included by incorporating a model of the spectral envelope in the DFT-based resampling scheme described earlier. Also, this pitch-shifting changes the duration of the signal, so an accompanying deletion or repetition of the resampled pitch periods is required to preserve the original time scale. Finally, given a pitch period segmentation of the signal, the signal can be viewed as the output of a time-varying source-filter model where the source is a pitch periodic impulse train and the time-varying filter determines the shape of the pitch period grains. In this light, a second time-varying pitch-synchronous filter can be applied to the signal by convolution with the individual pitch periods; the signal is then reconstructed by overlap-add of the new period segments. This notion leads to some time-varying modifications as well as pitch-based cross-synthesis of multiple signals.

As described in Section 2.7, signals with pitched behavior are well-suited for modification. The ease of modification based on the pitch-synchronous representation is thus not particularly surprising. As a final note, it should be clear that the PSR is not immediately useful for signal coding but that it does expose redundancies in the signal that can be exploited by further processing to achieve a compact representation. Two such processing techniques are described in the following sections.

## 5.3 Pitch-Synchronous Sinusoidal Models

The peak picking, line tracking, and phase interpolation problems in sinusoidal modeling can be resolved by applying Fourier methods to a resampled pitch-synchronous signal representation. These simplifications are a direct result of the prior effort put into pitch detection and signal segmentation. The pitch-synchronous representation is itself a signal-adaptive parametric model of the signal; by constructing the PSR, the signal is cast into a form which enables a Fourier expansion to be used in an effective manner.

Of course, it is commonplace to apply Fourier series to periodic signals; it indeed provides a compact representation for purely periodic signals. Here, the Fourier series approach is applied to pseudo-periodic signals on a period-by-period basis.

### 5.3.1 Fourier Series Representations

A detailed review of Fourier series methods is given in Appendix B; various connections between the DFT and expansions in terms of real sines and cosines are indicated there. The result that is of primary interest here is that a real signal of length $P$ can be expressed as

$$x[n] \;=\; \frac{X[0]}{P} \;+\; \frac{2}{P}\sum_{k} |X[k]| \cos\left(\omega_k n + \phi_k\right), \tag{5.3}$$

where $\omega_k = 2\pi k/P$, $|X[k]|$ and $\phi_k$ are respectively the magnitude and phase of the $k$-th bin of a size $P$ DFT of $x[n]$, and $k$ ranges over the half spectrum $[0, P/2]$. Note that this magnitude-phase form resembles the sinusoidal model of Chapter 2. The next section considers applying the representation of Equation (5.3) to the rows of a PSR, *i.e.* the pitch periods of a signal. This approach results in a pitch-synchronous sinusoidal model in which some of the difficulties of the general sinusoidal model are circumvented. The various simplifications arise because of the effort given to the process of pitch detection and signal segmentation.

### 5.3.2 Pitch-Synchronous Fourier Transforms

Applying the Fourier series to the pitch-synchronous representation of a signal is equivalent to carrying out pitch-synchronous sinusoidal modeling. In this case, as explained below, the peak picking and line tracking problems are eliminated by the pitch synchrony.

#### Peak picking

The DFT of a pitch period samples the DTFT at the frequencies of the pitch harmonics, namely the frequencies $\omega_k = 2\pi k/P$ for a pitch period of length $P$. These frequencies correspond to the relevant partials for the sinusoidal model. With regards to the discussion of Section 2.3.1, taking the DFT of a pitch period in the PSR is analogous to using a rectangular window that spans exactly one pitch period, which provides exact resolution of the harmonic components without spectral oversampling. In short, spectral peaks do not need to be sought out as in the general sinusoidal model; here, each of the spectral samples in the DFT corresponds directly to a partial of the signal model. Partials with small amplitude can be neglected in order to reduce the complexity of the model and the computation required for synthesis, but this may lead to discontinuities as discussed later.

#### Line tracking

In the pitch-synchronous sinusoidal model, the simplification of peak picking in the Fourier spectrum is accompanied by a simplification of the line tracking process. Indeed, no line tracking is necessary. A harmonic structure is imposed on the signal by the model, so the partial tracks are well-behaved by construction. Of course, it is necessary that the original signal exhibit pseudo-periodic behavior for this approach to be at all effective; given this foundation, the imposition of harmonic structure is by no means restrictive. Note that this insight applies to the case of a single note with an onset. To generate tracks that persist across multiple notes, it is necessary to either impose births and deaths in the transition regions or to carry out line tracking of the harmonics across the transitions.

### 5.3.3 Pitch-Synchronous Synthesis

Since it is a basis expansion, the Fourier series representation can achieve perfect reconstruction. Synthesis using basis vectors, however, is not particularly flexible. A generalized synthesis can be formalized by expressing a pitch period $x_i[n]$ in the magnitude-phase form of Equation (5.3) and then phrasing the synthesis as a sum-of-partials model. This framework is considered in the following sections.

#### Synthesis using a bank of oscillators

For a pitch period $x_i[n]$ of length $P$, the perfect reconstruction magnitude-phase expression is given by

$$x_i[n] \;=\; \frac{2}{P}\sum_k |X_i[k]| \cos\left(\omega_k n + \phi_{k,i}\right) \tag{5.4}$$

for $n \in [0, P-1]$ and $\omega_k = 2\pi k/P$. The signal can be constructed by concatenating the pitch period frames:

$$x[n] \;=\; \sum_i x_i[n] \;=\; \frac{2}{P}\sum_i \sum_k |X_i[k]| \cos\left(\omega_k n + \phi_{k,i}\right), \tag{5.5}$$

where $i$ is a frame index; $x_i[n]$ is the $i$-th frame, and $X_i[k]$ is the DFT of $x_i[n]$. The segment $x_i[n]$ is supported on the interval $n \in [iP, iP + P - 1]$; $X_i[k]$ likewise corresponds to that time interval. More formally, this Fourier amplitude could be expressed as

$$X_i[k] \left( u[n - iP] - u[n - (i+1)P] \right), \tag{5.6}$$

where $u[n]$ is the unit step function; the simpler notation is adhered to in this treatment.

While the same frequencies appear in the model of Equation (5.5) in every frame, there are not necessarily actual partials that persist smoothly in time. Consider the contribution of the components at a single frequency $\omega_k$:

$$p_k[n] \;=\; \frac{2}{P} \sum_i |X_i[k]| \cos\left(\omega_k n + \phi_{k,i}\right). \tag{5.7}$$

The phase terms are not necessarily the same in each frame, so for this single-frequency component the concatenation may have discontinuities at the frame boundaries. These discontinuities are eliminated in the full synthesis; their appearance in the constituent signals, however, indicates that if components are omitted to achieve compaction, frame-rate discontinuities will appear in the output. Because of these discontinuities, the Fourier model in Equation (5.5) cannot be simply interpreted as a sum of partials.

The difficulty with phase discontinuities at the frame boundaries can be circumvented by rephrasing the reconstruction as a sinusoidal synthesis using a bank of oscillators. Rather than relying on the standard Fourier basis functions, sinusoidal expansion functions that interpolate the amplitude and phase are generated such that the reconstruction indeed consists of evolving partials and not discrete Fourier atoms with the aforementioned boundary mismatches caused by phase misalignment. This revision of the approach provides an example of how a parametric model can improve compaction: in the approximate reconstructions of compressed models, discontinuities occur at the frame boundaries in the basis case but not in the sinusoidal synthesis; the sinusoidal model is free from boundary discontinuities by construction. Note however that this sinusoidal model, while it is perceptually accurate, does not carry out perfect reconstruction.

### Zero-phase sinusoidal modeling

In the standard sinusoidal model, the phase interpolation process at the synthesis stage is a high-complexity operation. Phase interpolation is thus one of the major obstacles in achieving real-time synthesis [132]. This difficulty is circumvented here by imposing a harmonic structure via the processes of pitch detection, segmentation, and resampling.

In the pitch-synchronous sinusoidal model introduced above, the phase of the harmonics is preserved; phase interpolation from frame to frame is thus required, but this is problematic in several respects. First, it is computationally expensive. Second, the interpolation does not take into account a fundamental property of the representation, namely that the same frequencies are present in every frame; indeed, by fitting a cubic polynomial to the frequency and phase parameters in adjacent frames, the effective frequency will be time-varying, which is not desired in this pitch-synchronous algorithm.

By construction, a Fourier sinusoid in a frame moves through an integral number of periods, meaning that its start and end phases are the same (one sample off, that is). Thus, for the corresponding sinusoid in the next frame to evolve continuously across the frame boundary, its starting phase should be one sample ahead of the end phase in the previous frame, or in other words it should be equal to the start phase from the previous frame. If this continuity is imposed, there is no phase interpolation required in the synthesis; a harmonic partial has the same phase in every frame.

This method is referred to here as *zero-phase* sinusoidal modeling since the start phases in the first frame can all be set to zero; then, the start phase for every partial in every frame is zero. In some cases, it may be useful to preserve the phase in the first frame to ensure perfect reconstruction there; this technique can be used to reconstruct attacks without the delocalization incurred in the general sinusoidal model. This initial phase is then fixed as the start phase for all frames, so the signal reconstruction can be phrased as

$$\hat{x}[n] \;=\; \sum_i x_i[n] \;=\; \frac{2}{P}\sum_i\sum_k |X_i[k]|\cos\left(\omega_k n + \phi_{k,0}\right) \tag{5.8}$$

$$=\; \frac{2}{P}\sum_k \cos\left(\omega_k n + \phi_{k,0}\right)\sum_i |X_i[k]| \tag{5.9}$$

$$=\; \sum_k \cos\left(\omega_k n + \phi_{k,0}\right)\sum_i A_{k,i}[n], \tag{5.10}$$

where the $A_{k,i}[n]$ are stepwise amplitude parameters that correspond directly to the Fourier coefficients:

$$A_{k,i}[n] \;=\; \frac{2}{P}|X_i[k]| \tag{5.11}$$

for $n \in [iP, iP + P - 1]$. Interpolation can be included to smooth the stepwise amplitude envelopes of the partials in the reconstruction. Then, the signal model is:

$$\hat{x}[n] \;=\; \sum_i\sum_k A_{k,i}[n]\cos\left(\omega_k n + \phi_{k,0}\right) \;=\; \sum_k \cos\left(\omega_k n + \phi_{k,0}\right)\sum_i A_{k,i}[n], \tag{5.12}$$

which is simply a sum of partials with constant frequencies $\omega_k$, each modulated by a linear amplitude envelope given by

$$A_{k,i}[n] \;=\; \frac{2}{P}\left[\frac{nX_i[k] + (P-n)X_{i-1}[k]}{P}\right]. \tag{5.13}$$

In the first frame, where $i = 0$, the amplitude envelope is defined as a constant

$$A_{k,0}[n] \;=\; \frac{2}{P}X_0[k] \qquad n \in [0, P-1] \tag{5.14}$$

so that perfect reconstruction is carried out there. More generally, this perfect reconstruction can be carried out over an arbitrary number of frames at the onset to represent the transient accurately. Recalling from the discussion of Section 5.2.3 that the prototypical signal consists of an unpitched region followed by a pitched region, the approach is to model the entire unpitched region perfectly in the above fashion; once the pitched region is entered, the phase is fixed and the harmonic sum-of-partials model of Equation (5.12) is used.

Many variations of pitch-synchronous Fourier series modeling can be formulated. For instance, the amplitude interpolation can be carried out between the centers of adjacent pitch period frames rather than between the frame boundaries; this is similar to the way the synthesis frames in the sinusoidal model are defined between the centers of the analysis frames. Such variations will not be considered here; some related efforts involving zero-phase modeling, or magnitude-only reconstruction, have been discussed in the literature [149, 204]. The intent here is primarily to motivate the usefulness of parametric analysis and adaptivity for signal modeling; estimating the pitch parameter leads to simple sinusoidal models, and incorporating perfect reconstruction allows for accurate representation of transients. Note that in either zero-phase or fixed-phase modeling, the elimination of the phase information results in immediate data reduction, and that this compression is transparent since it relies on the well-known principle that the ear is insensitive to the relative phases of component signals.

### 5.3.4 Coding and Modification

There is a substantial amount of redundancy from one pitch period to the next; adjacent periods of a signal have a similar structure. This self-similarity is clearly depicted in the pitch-synchronous representation shown in Figure 5.2 and is of course the fundamental motivation for pitch-synchronous processing. Since adjacent periods are redundant or similar, the expansion coefficients of adjacent periods exhibit a corresponding similarity. Because of this frame-to-frame dependence, the expansion coefficients can be subsampled and/or coded differentially. Furthermore, multiresolution modeling can be carried out by subsampling the tracks of the low frequency harmonics more than those of the high frequency ones; such subsampling reduces both the model data and the amount of computation required for synthesis. Indeed, the tracks can be approximated in a very rough sense; variations between pitch periods, which may be important for realism, can be reincorporated in the synthesis based on simple stochastic models.

The signal modifications discussed in Section 2.7 can all be carried out in the pitch-synchronous sinusoidal model. It is interesting to note that some modifications such as time-scaling and pitch-shifting can either be implemented based on the sinusoidal parameterization or via the granular model of the PSR matrix. Note that modifications which involve resampling are accelerated in the pitch-synchronous sinusoidal model because the Fourier series representation can directly be used for resampling as described in Section 5.2.2.

## 5.4 Pitch-Synchronous Wavelet Transforms

This section considers applying the wavelet transform in a pitch-synchronous fashion as originally proposed in [205, 195]. The pitch-synchronous wavelet transform (PSWT) is developed as an extension of the wavelet transform that is suitable for pseudo-periodic signals; the underlying signal models are discussed for both cases. After the algorithm is introduced, implementation frameworks and applications are considered.

### 5.4.1 Spectral Interpretations

The wavelet transform and the pitch-synchronous wavelet transform can be understood most simply in terms of their frequency-domain operation. The spectral decompositions of each transform are described below.

#### The discrete wavelet transform

As discussed in Section 3.2.1, the signal model underlying the discrete wavelet transform (DWT) can be interpreted in two complementary ways. At the atomic level, the signal is represented as the sum of atoms of various scales; the scale is long in time at low frequencies and short for high frequencies. Each of these atoms corresponds to a tile in the time-frequency tiling given in Figure 1.9(b) in Section 1.5.2. This atomic or tile-based perspective corresponds to interpreting the discrete wavelet transform as a basis expansion; each atom or tile is a basis function.

Alternatively to the atomic interpretation, the wavelet transform can be thought of as an octave-band filter bank. As reviewed in Section 3.2.1, the discrete wavelet transform can be implemented with a critically sampled perfect reconstruction filter bank with a general octave-band structure; the coefficients of the atomic signal expansion in a wavelet basis can be computed with such a filter bank. This filter bank equivalence is clearly evident in the tiling diagram of Figure 1.9(b); considered across frequency, the structure of the tiles indicates an octave-band demarcation of the time-frequency plane. These bands in the tiling correspond to the subbands of the wavelet filter bank; in frequency, then, a wavelet filter bank splits a signal into octave bands, plus a final lowpass band. In a tree-structured iterated filter bank implementation, this final lowpass band corresponds to the lowpass branch of the final iterated stage; this branch is of particular interest for signal coding since it is highly downsampled.

The filter bank interpretation shows that the discrete wavelet transform provides a signal model in terms of octave bands plus a final lowpass band. The lowpass band is a coarse estimate of the signal. The octave bands provide details that can be added to successively refine the signal; perfect reconstruction is achieved if all of the subbands are included. This lowpass-plus-details model is appropriate for signals which are primarily lowpass; the wavelet transform has thus been applied successfully in image compression [18, 19]. However, for signals with wideband spectral content, such as high-quality audio, a lowpass estimate is a poor approximation. For any pseudo-periodic signals with high-frequency harmonic content, a lowpass estimate does not incorporate the high-frequency harmonics. Indeed, for general wavelet filter banks based on lowpass-highpass filtering at each iteration, representing a signal in terms of the final lowpass band simply amounts to lowpass filtering the signal and using a lower sampling frequency, so it is not surprising that this compaction approach does not typically provide high-quality audio. Wavelet-based modeling of a bassoon signal is considered in Figure 5.3 for the case of Daubechies wavelets of length eight; these wavelets will be used for all of the simulations in this chapter. Given the modeling inadequacy indicated in Figure 5.3, it is of interest to adjust the wavelet transform so that the signal estimate includes the higher harmonics. This adjustment is arrived at via the following consideration of upsampled wavelets.

### Upsampled wavelets

As motivated in the previous section, it is of interest to modify the wavelet transform in such a way that the coarse estimate of a pseudo-periodic signal includes the signal harmonics. Conceptually, the first step in achieving this spectral revision is to consider the effect of upsampling the impulse responses of the iterated filters in a wavelet analysis-synthesis filter bank. The spectral motivation is described after the following mathematical treatment.

As derived in Appendix A, a wavelet filter bank can be constructed by iterating critically sampled two-channel filter banks that satisfy the perfect reconstruction condition

$$\begin{aligned} G_0(z)H_0(z) \;+\; G_1(z)H_1(z) \;&=\; 2 \qquad (5.15)\\ G_0(z)H_0(-z) \;+\; G_1(z)H_1(-z) \;&=\; 0, \qquad (5.16)\end{aligned}$$

where the $H_i(z)$ are the analysis filters and the $G_i(z)$ are the synthesis filters. Note that the condition still holds if the transformation $z \rightarrow z^M$ is carried out:

$$\begin{aligned} G_0(z^M)H_0(z^M) \;+\; G_1(z^M)H_1(z^M) \;&=\; 2 \qquad (5.17)\\ G_0(z^M)H_0(-z^M) \;+\; G_1(z^M)H_1(-z^M) \;&=\; 0. \qquad (5.18)\end{aligned}$$

As will be shown below, this transformed expression is not the same perfect reconstruction condition that arises if the constituent filters are upsampled; a comparison of the

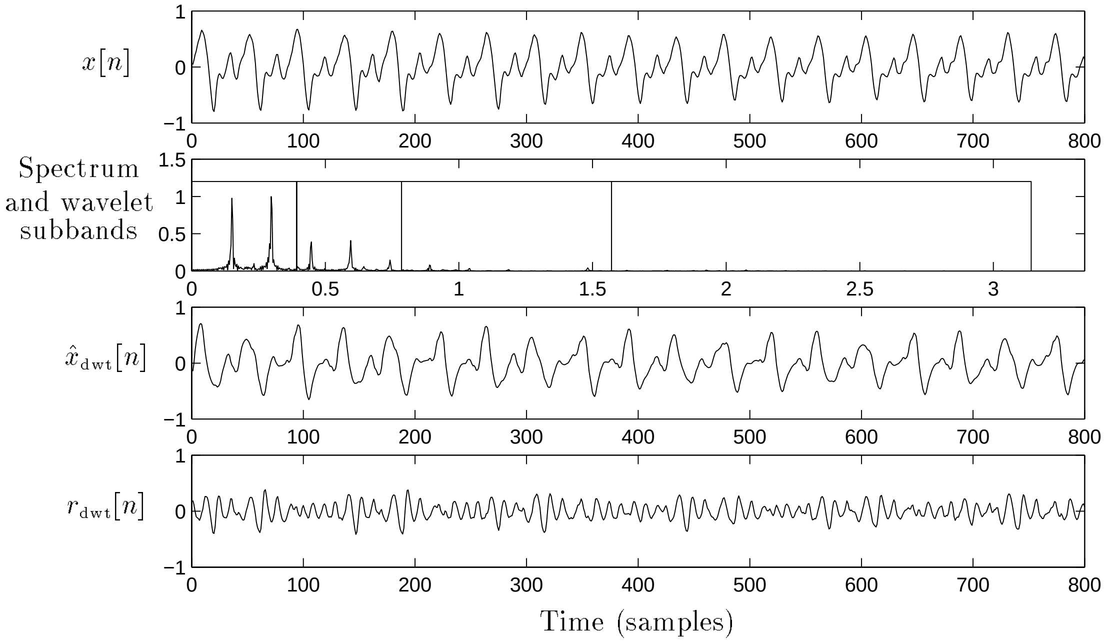

FIGURE 5.3: The discrete wavelet transform provides an octave-band decomposition of a signal. Compaction is achieved by representing the signal in terms of the highly downsampled lowpass band; the estimate can be successively refined by incorporating the octave-band details. For a wideband pitched audio signal such as the bassoon note shown, the higher harmonics extend throughout the wavelet subbands as indicated in the plot of the spectrum. The lowpass estimate $\hat{x}_{\mathrm{dwt}}[n]$ does not capture the signal behavior accurately. The residual $r_{\mathrm{dwt}}[n]$ is the sum of the octave-band details.

two expressions will lead to a simple sufficient condition for perfect reconstruction in an upsampled wavelet filter bank.

Given perfect reconstruction filters $\{G_0(z), G_1(z), H_0(z), H_1(z)\}$, the question at hand is whether the upsampled filters

$$\begin{aligned} A_0(z) &= G_0(z^M) & B_0(z) &= H_0(z^M) \\ A_1(z) &= G_1(z^M) & B_1(z) &= H_1(z^M) \end{aligned} \tag{5.19}$$

also provide perfect reconstruction in a two-channel filter bank. The constraint on the new filters is then

$$A_0(z)B_0(z) + A_1(z)B_1(z) = 2 \tag{5.20}$$

$$A_0(z)B_0(-z) + A_1(z)B_1(-z) = 0, \tag{5.21}$$

which can be readily expressed in terms of the original filters as

$$G_0(z^M)H_0(z^M) + G_1(z^M)H_1(z^M) = 2 \tag{5.22}$$

$$G_0(z^M)H_0((-1)^M z^M) + G_1(z^M)H_1((-1)^M z^M) = 0. \tag{5.23}$$

Comparing this to the expressions in Equations (5.17) and (5.18) indicates immediately that perfect reconstruction holds when $M$ is odd. An odd upsampling factor is thus *sufficient* but not necessary for perfect reconstruction, meaning that for some filters, an even $M$ will work, and for others not. The difficulty with an even $M$ can be readily exemplified for the case of the one-scale Haar basis depicted in Figure 5.4(a). Upsampling the underlying Haar wavelet filters by a factor of two yields the expansion functions shown in Figure 5.4(b), which clearly do not span the signal space and are thus not a basis. As a result, perfect reconstruction cannot be achieved with filters based on Haar wavelets upsampled by even factors.

By upsampling the wavelet filters, the spectral decomposition derived by the filter bank can be adjusted. The frequency-domain effect of upsampling is a compression of the spectrum by the upsampling factor, which admits spectral images into the range $[0, 2\pi]$. The subband of a branch in the upsampled filter bank then includes both the original band and these images. This spectral decomposition is depicted in Figure 5.5 for the case of a depth-three wavelet filter bank and upsampling by factors of three and nine. Whereas in the original wavelet transform the signal estimate is a lowpass version, in the upsampled transform the estimate consists of disparate frequency bands as indicated by the shading. The insight here is that upsampling of the filters can be used to redistribute the subbands across the spectrum so as to change the frequency-domain regions that the filter bank focuses on. As will be seen, such redistribution can be particularly effective for spectra with strong harmonic behavior, *i.e.* pseudo-periodic signals.

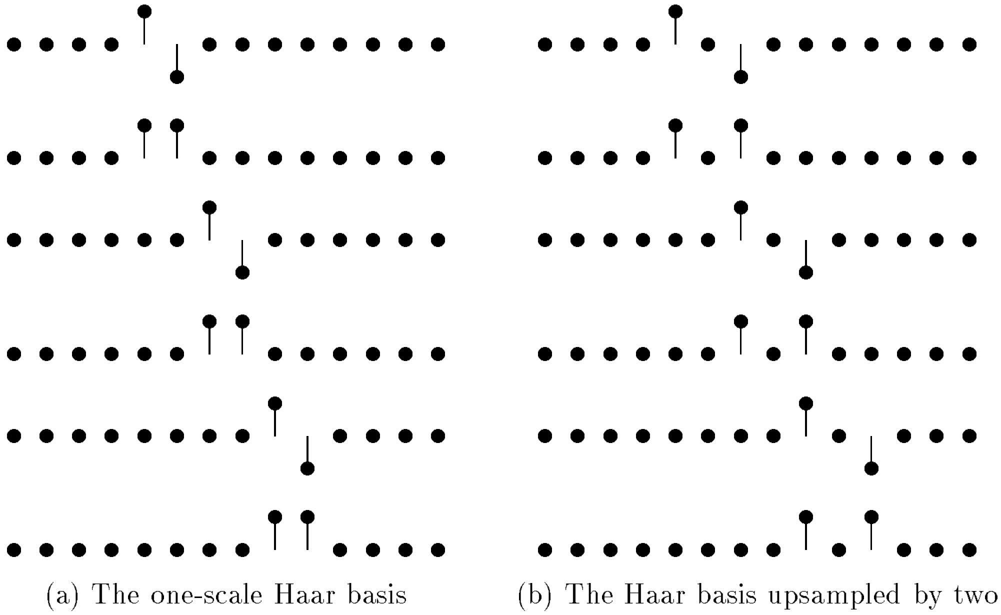
(a) The one-scale Haar basis (b) The Haar basis upsampled by two

FIGURE 5.4: The one-scale Haar basis shown in (a) is upsampled by two to derive the functions shown in (b), which clearly do not span the signal space.

Several issues about the upsampled wavelet transform deserve mention. For one, a model in terms of the lowpass wavelet subband and a model in terms of the upsampled lowpass band have the same amount of data. The upsampled case, however, differs from the standard case in that there is no meaningful tiling that can be associated with it because of the effect of the upsampling on the time-localization of the atoms in the decomposition. In a sense, the localizations in time and frequency are both banded, but this does not easily lend itself to a tile-based depiction. For this reason, the upsampled wavelet transform and likewise the pitch-synchronous wavelet transform to be discussed cannot be readily interpreted as an atomic decomposition. The granularity of the PSWT arises from the pitch period segmentation and not from the filtering process.

### Pitch-period upsampling

For signals with wideband harmonic structure, the lowpass estimate of the wavelet signal model does not accurately represent the signal. In the previous section, it was shown that upsampling the wavelet filters adjusts the spectral decomposition derived by the filter bank. If the wavelets in the filter bank are upsampled by the pitch period, the result is that the lowpass band is reallocated in the spectrum to the regions around the harmonics. The upsampled filter has a passband at each harmonic frequency; the disparate bands are indeed coupled. The subband signal of the harmonic band provides a pseudo-periodic estimate of the signal rather than a lowpass estimate. This leads to the periodic-plus-details

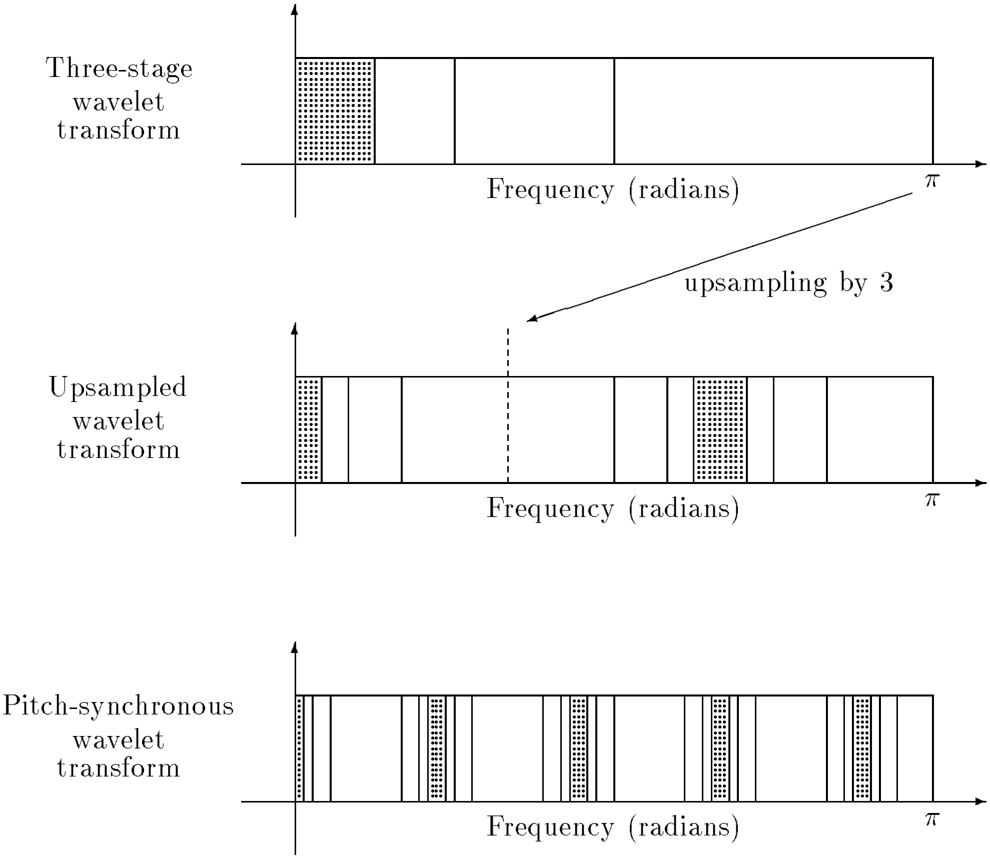
Three-stage wavelet transform
Frequency (radians)
π
upsampling by 3
Upsampled wavelet transform
Frequency (radians)
π
Pitch-synchronous wavelet transform
Frequency (radians)
π

Figure 5.5: The spectral decompositions of a wavelet transform of depth three and the corresponding upsampled wavelet transform for an upsampling factor of three are given in the first two diagrams. The shaded regions correspond to the lowest branches of the transform filter bank trees, which generally provide the signal estimates. A higher degree of upsampling (9) yields the decomposition in the third plot. Such a decomposition is useful if the signal has harmonics that fall within the harmonically-spaced shaded bands; such structure can be imposed by using the pitch period as the upsampling factor. Note that only the positive-frequency half-band is shown in the plots.

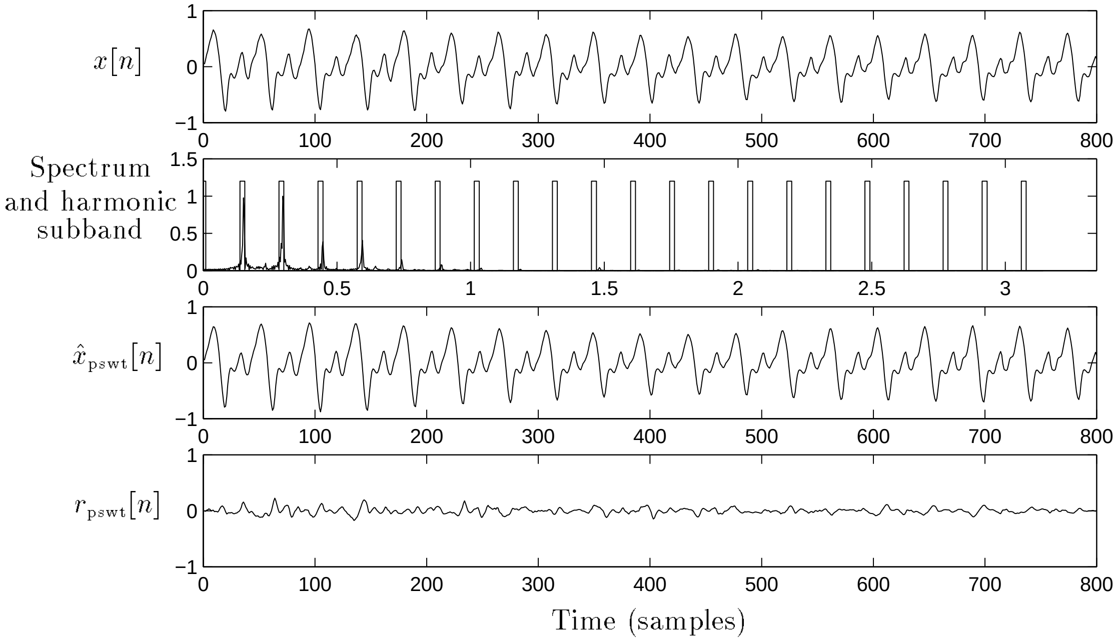

FIGURE 5.6: The pitch-synchronous wavelet transform provides a decomposition of the signal localized around the harmonic frequencies. Compaction is achieved by representing the signal in terms of the narrow bands around the harmonics, which are coupled into one subband in the PSWT; the estimate can be refined by incorporating the inter-harmonic details. The inter-harmonic bands are not shown in the spectral plot for the sake of neatness. For a wideband pitched audio signal such as the bassoon note shown, the harmonic estimate $\hat{x}_{\text{pswt}}[n]$ captures the signal behavior much more accurately than the lowpass estimate of the wavelet transform, namely the signal $\hat{x}_{\text{dwt}}[n]$ plotted in Figure 5.3. The residual $r_{\text{pswt}}[n]$ is the sum of the inter-harmonic details, and is clearly of lower energy than the wavelet residual $r_{\text{dwt}}[n]$ in Figure 5.3.

signal model of the pitch-synchronous wavelet transform. A depiction of the spectral decomposition of the PSWT is shown in Figure 5.5; the harmonic band indicated by the shaded regions provides the pseudo-periodic signal estimate, and the inter-harmonic bands derive the detail signals. The estimate is a version of the signal in which local period-to-period variations have been removed; these variations are represented by the detail signals, and can be incorporated in the synthesis if needed for perceptual realism.

An example of the PSWT signal model is given in 5.6. It should be noted that the same amount of data is involved in the PSWT signal model of Figure 5.6 and the DWT signal model of Figure 5.3. The harmonic band of the PSWT simply captures the signal behavior more accurately than the lowpass band of the DWT. Implementation of the PSWT is discussed in the next section; because of the problem associated with upsampling by even factors, other methods of generating the harmonic spectral decomposition are considered.

### 5.4.2 Implementation Frameworks

The pitch-synchronous wavelet transform can be implemented in a number of ways. These are described below; the actual expansion functions in the various approaches are rigorously formalized in [205, 195].

#### Comb wavelets

Based on the discussion on the spectral effect of upsampling a wavelet filter bank, a direct implementation of a pitch-synchronous wavelet transform simply involves upsampling by the pitch period $P$. The corresponding spectral decomposition has bands centered at the harmonic frequencies, and the signal is modeled in a periodic-plus-details fashion as desired. An important caveat to note, however, is that these *comb wavelets*, as derived in the treatment of upsampled wavelets and illustrated for the simple Haar case, do not guarantee perfect reconstruction if $P$ is even. Because of this limitation, it is necessary to consider other structures that arrive at the same general spectral decomposition.

#### The multiplexed wavelet transform

The problem with the spanning space in the case of comb wavelets can be overcome by using the multiplexed wavelet transform depicted in Figure 5.7. Here, the signal is demultiplexed into $P$ subsignals, each of which is processed by a wavelet transform; these $P$ subsignals correspond to the columns of the PSR matrix. The lowpass estimate in the wavelet transform of a subsignal is then simply a lowpass version of the corresponding PSR column. A pseudo-periodic signal estimate can be arrived at by reconstructing a PSR matrix using only the lowpass signals and then concatenating the rows of the matrix. The net effect is that of pitch-synchronous filtering: period-to-period changes are filtered out. Perfect reconstruction can be achieved by incorporating all of the subband signals of each wavelet transform.

#### Interpretation as a polyphase structure

Polyphase methods have been of some interest in the literature, primarily as a tool for analyzing filter banks [2]. Here, it is noted that the multiplexed wavelet transform described above can be interpreted as a *polyphase* transform; a block diagram is given in Figure 5.8. The term polyphase simply means that a signal is treated in terms of progressively delayed and subsampled components, *i.e.* the phases of the signal. In the pitch-synchronous case, the signal is modeled as having $P$ phases corresponding to the $P$ pitch-synchronous subsignals.

In Figure 5.8, the subsignals are processed with a general transform $T$. For the pitch-synchronous wavelet transform, this should obviously be a wavelet transform. If

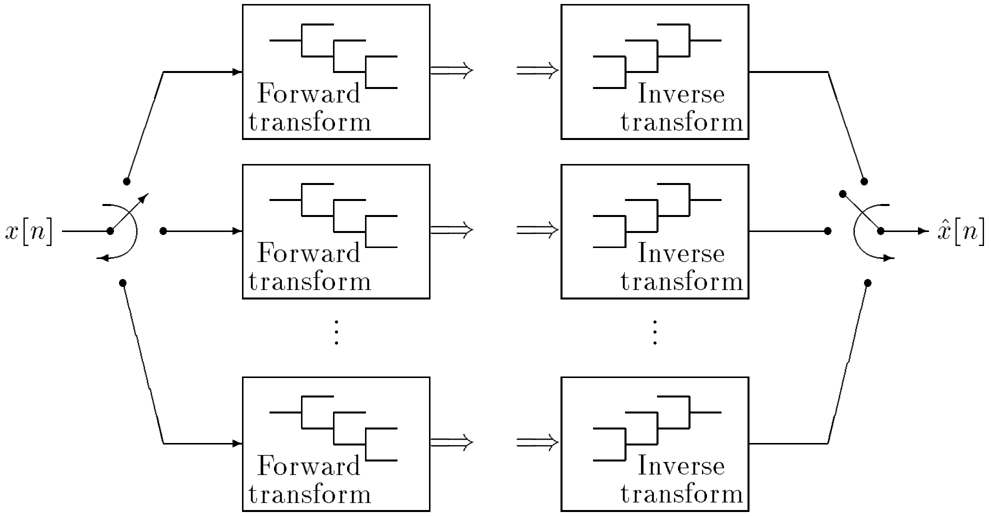

FIGURE 5.7: Schematic of the multiplexed wavelet transform. If the number of branches is equal to the number of samples in a pitch period, this structure implements a a pitch-synchronous wavelet transform.

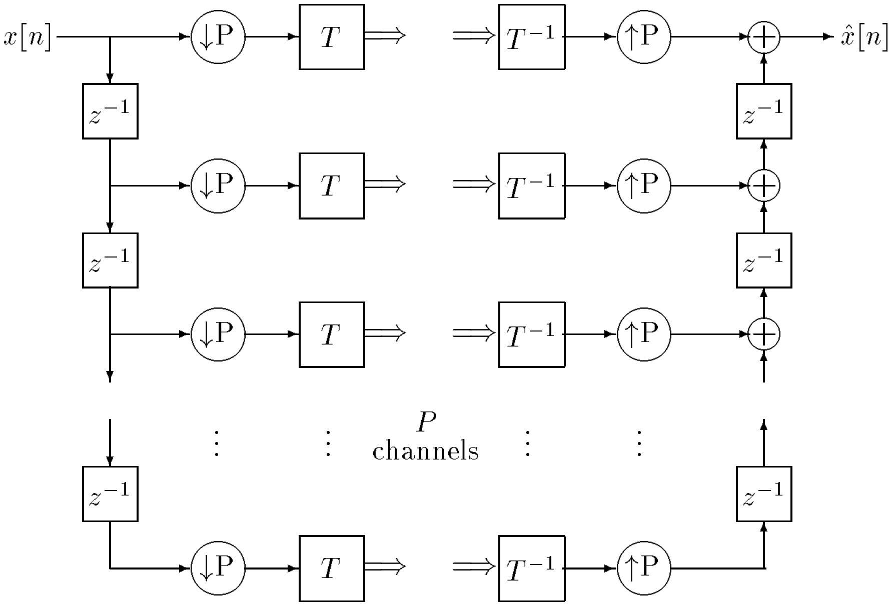

FIGURE 5.8: Polyphase formulation of the pitch-synchronous wavelet transform. Perfect reconstruction holds for the entire system if the channel transforms provide perfect reconstruction. This structure is useful for approximating a signal with period $P$; the overall signal estimate is pseudo-periodic if the channel transforms provide lowpass estimates of the polyphase components.

only the lowpass bands of the wavelet transforms are retained, the signal reconstruction is a pseudo-periodic estimate of the original signal. Indeed, such an estimate can be arrived at by applying any type of lowpass filters in the subbands; this structure is by no means restricted to wavelet transforms. For nonstationary or arbitrary signals it may even be of interest to consider more general transforms, and perhaps joint adaptive optimization of $P$ and the channel transforms.

#### Two-dimensional wavelet transforms

The pitch-synchronous wavelet transform takes advantage of the similarity between adjacent pitch periods by carrying out a wavelet transform on the columns of the PSR matrix. Typical signals, however, also exhibit some redundancy from sample to sample; this redundancy is not exploited in the PSWT, but is central to the DWT. To account for both types of redundancy, the PSR can be processed by a two-dimensional wavelet transform; for separable two-dimensional wavelets, this amounts to coupling the PSWT and the DWT. A similar approach has been applied successfully to ECG data compression [37]. It is an open question, however, if this method can be used for high-quality compression of speech or audio.

### 5.4.3 Coding and Modification

In this section, applications of the pitch-synchronous wavelet transform for signal coding and modification are considered. In the pitch-synchronous sinusoidal model, modifications were enabled both by the granularity of the representation and its parametric nature; here, the modifications based on granulation are still applicable. However, the pitch-synchronous wavelet transform does not readily support additional modifications; for instance, modification of the spectral components leads to discontinuities in the reconstruction as will be shown below. After a discussion of modifications, coding issues are explored. The model provides an accurate and compact signal estimate for pitched signals; furthermore, transients can also be accurately modeled since the transform is capable of perfect reconstruction.

#### Spectral shaping

In audio processing and especially computer music, novel modifications are of great interest. An immediate modification suggested by the spectral decomposition of the pitch-synchronous wavelet transform is that of spectral shaping. If gains are applied to the subbands, the spectrum can seemingly be reshaped in various ways to achieve such modifications. However, this approach has a subtle difficulty similar to the problem in the discrete wavelet transform wherein the reconstruction and aliasing cancellation constraints

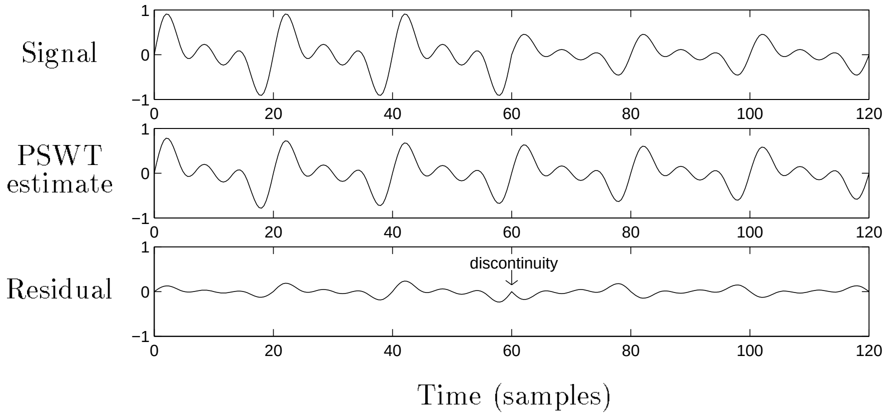

Figure 5.9: The pitch-synchronous wavelet transform provides a signal decomposition in terms of a pseudo-periodic estimate and detail signals. The model shown here results from a three-stage transform; the residual is the sum of the details. The discontinuity occurs because the estimate is subsequently greater than and less than the original signal in adjacent periods.

are violated if the subbands are modified. The problem can be easily understood by considering the signal model and the averaging process. The pseudo-periodic original signal may exhibit amplitude variations from period to period. The estimate signal is derived by averaging these varying pitch period signals; by the nature of averaging, sometimes the estimate will be greater than the original and sometimes less. Such a transition is shown in Figure 5.9. The model residual, which is the sum of the detail signals, exhibits discontinuities at these transition points. These discontinuities cancel in a perfect signal reconstruction; if the subbands are modified, however, the discontinuities may appear in the synthesis. This lack of robustness limits the usefulness of spectral manipulations in the PSWT signal model. Given the reconstruction difficulties, other modifications are basically restricted to those that result from the granularity of the pitch-synchronous representation; these were discussed in Section 5.2.4

### Signal estimation

In the discrete wavelet transform, the lowpass branch provides a coarse estimate of the signal; discarding the other subbands yields a compact model since the lowpass branch is highly downsampled. This type of modeling has proven quite useful for image coding [18, 19]. For audio, however, a lowpass estimate neglects high frequency content and thus tends to yield a low-quality reconstruction. Building from this observation, the pitch-synchronous wavelet transform estimates the signal in terms of its spectral content around its harmonic frequencies. For a pitched signal, these are the most active regions in the spectrum and thus the PSWT estimate captures more of a pitched signal's behavior than the DWT. Figures 5.3 and 5.6 can be compared to indicate the relative performance of the DWT and the PSWT for modeling or estimation of pitched signals.

### Coding gain

A full treatment of the multiplexed wavelet transform for signal coding is given in [195, 206]. The fundamental reason for the coding gain is that the periodic-plus-details signal model is much more appropriate for signals with pseudo-periodic behavior than standard lowpass-plus-details models. For the same amount of model data, the PSWT model is more accurate than the DWT model. In rate-distortion terminology, the PSWT model has less distortion than the DWT model at low rates; at high rates, where more subbands are included to refine the models, the distortion performance is more competitive. One caveat in this comparison is that if the original signal varies in pitch in a perceptually meaningful way, it is necessary to store the pitch period values as side information so that the output pitch periods can be resampled at the synthesis stage; in cases where the pitch variations are important and must be reintroduced, some overhead is required. Such coding issues are not considered in further depth. The next two sections deal with relevant modeling issues that arise in the pitch-synchronous wavelet transform; where appropriate, comments on signal coding are given.

### Stochastic models of the detail signals

Signal estimates based on the pitch-synchronous wavelet transform are shown in Figures 5.6 and 5.9. These estimates are smooth functions that capture the key musical features of the original signals such as the pitch and the harmonic structure. The PSWT estimate is thus analogous to the deterministic component of the sinusoidal model; in both cases, the decompositions directly involve the spectral peaks. Similarly, the detail signals have a correspondence to the stochastic component of the sinusoidal model. Given this observation, it is reasonable to consider modeling the detail signals as a noiselike residual. Such approaches are discussed in [195]. This analogy between the sinusoidal model and the PSWT, of course, is limited to pseudo-periodic signals; for signals consisting of evolving harmonics plus noise, the deterministic-plus-stochastic (*i.e.* reconstruction-plus-residual) models are similar.

### Pre-echo in the reconstruction

Like the other signal models presented in this thesis, models based on the wavelet transform can exhibit pre-echo distortion. Of course, this pre-echo is not introduced if perfect reconstruction is carried out; the problem arises when the signal is modeled in terms of the lowpass subband, which cannot accurately represent transient events. This distortion is considered here for the case of the discrete wavelet transform first. Then,

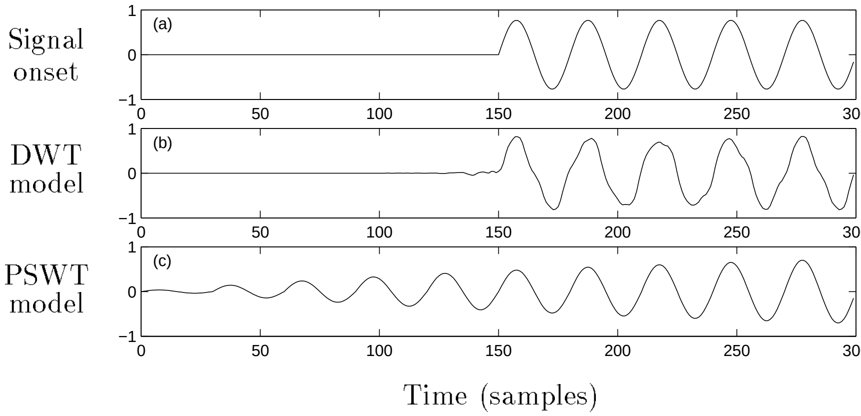

FIGURE 5.10: Pre-echo is introduced in the discrete wavelet transform when a transient signal (a) is estimated in terms of the lowpass subband (b). The pre-echo is significantly increased in the pitch-synchronous wavelet transform model (c) since the discrete wavelet transform pre-echo occurs in each of the subsignals; noting the structure of Figure 5.8, the pre-echo in each subsignal is upsampled by the pitch period in the reconstruction, which accounts for the spreading.

by considering the pre-echo in the DWT models of the pitch-synchronous subsignals, it is shown that the pre-echo problem is more severe in the pitch-synchronous wavelet transform.

As discussed in Section 3.2.1, the lowpass subband in the discrete wavelet transform is characterized by good frequency resolution and poor time resolution. The result is that transients are delocalized in signal estimates based on the lowpass subband. Consider the signal onset in Figure 5.10(a) and its lowpass wavelet model shown in Figure 5.10(b). Pre-echo is introduced in the lowpass model of the onset. Note that some of this pre-echo actually results from precision effects in the wavelet filter specification, but that the majority of it is caused by the poor time localization of the low-frequency subband.

In the pitch-synchronous wavelet transform, each of the subsignals is modeled in terms of the lowpass band of a discrete wavelet transform, which means that each of these downsampled signals is susceptible to the pre-echo of the DWT estimation process. The subsignal pre-echo occurs in the time domain at a subsampled rate, specifically a factor $P$ less than the original signal as indicated in the block diagram of Figure 5.8. At the synthesis side, the subsignals are upsampled by the pitch period $P$ and as a result the pre-echo is spread out by a factor of $P$. This drastic increase in the pre-echo is illustrated in Figure 5.10(c). Another example of PSWT signal estimation and pre-echo is given in Figure 5.11; in this case, the DWT clearly provides a poor model when compared to that given by a PSWT involving the same amount of model data, *i.e.* the same depth of filtering. This example is discussed further in the next section.

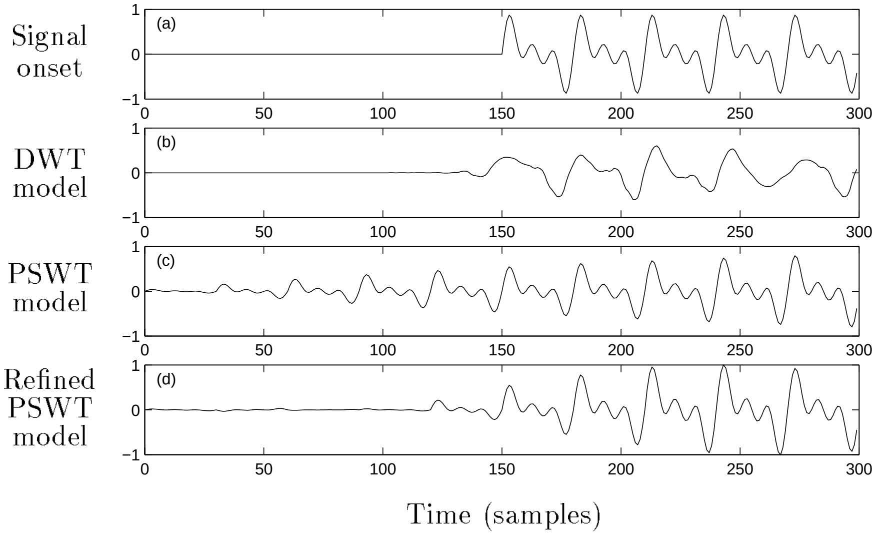

Figure 5.11: Pre-echo in models of a synthetic signal (a) with higher harmonics. Though the two models involve the same amount of data, the lowpass DWT model in (b) is clearly a much less accurate signal estimate than the PSWT model in (c). In the PSWT, however, the pre-echo is spread out by a factor equal to the pitch period. By incorporating the detail signals in the onset region, the pre-echo can be reduced; perfect reconstruction of the transient is achieved by adding all of the details in the vicinity of the onset.

### Perfect reconstruction of transients

In the previous section, it was shown that pre-echo occurs in compact PSWT models of transient signals. Indeed, pre-echo is a basic problem in compact models; here, the problem can be readily solved since the PSWT is capable of perfect reconstruction. The idea is to only use the compact model where appropriate and to carry out perfect reconstruction where the compact model is inaccurate, namely near the transients. Near transients, there is significant energy in the detail signals of the PSWT; if this condition occurs then the subbands should be included in the model. In terms of the PSR matrix, this corresponds to representing the first few rows of the matrix exactly. Once the periodicity of the signal is established, most of the energy falls in the harmonic bands and the inter-harmonic bands can be discarded without degrading the estimate. Thus, compaction is achieved in the pitched regions but not in the unpitched regions. An example of this signal-adaptive representation is given in Figure 5.11, which shows the pre-echo reduction that results from incorporating one detail signal into the reconstruction. With the exception of filter precision effects, perfect reconstruction is achieved if all of the detail signals are included; inclusion of all the details is generally desired so as to avoid introducing the aforementioned discontinuities into the reconstruction.

In coding applications, the additional cost of allowing for perfect reconstruction of transients is not significant; in a musical note, for instance, the attack is typically much shorter than the pseudo-periodic sustain region, so perfect reconstruction is required only over a small percentage of the signal. Furthermore, since the attack region is perceptually important, perfect reconstruction of the attack is worthwhile from the standpoint of psychoacoustics; transparent modeling of attacks is necessary for high-quality audio synthesis. In application to musical phrases, then, perfect reconstruction is carried out in the unpitched regions while harmonic PSWT modeling is carried out for the pitched regions. This process preserves note transitions. In a sense, it also introduces a concurrency in the unpitched regions similar to that of the deterministic-plus-stochastic model. When the signal exhibits transient behavior, a full model with concurrent harmonics and inter-harmonic details is used, whereas in stable pitched regions, the harmonic model alone is used.

This approach of signal-adaptive modeling and reconstruction in the PSWT can be interpreted as a filter bank where only subbands with significant energy are included in the synthesis. Similar ideas have been employed in compression algorithms based on more standard filter bank structures such as the discrete wavelet transform and uniform filter banks [2, 20].

## 5.5 Applications

Of course, pitch-synchronous methods such as the ones discussed in this chapter have immediate applications in audio processing. These have been considered throughout the chapter; a few further issues are treated in Section 5.5.1. Pitch-synchronous methods can also be applied to any signals with pseudo-periodic behavior, *e.g.* heartbeat signals. The advantages of any such methods result from the effort applied to estimation of the pitch parameter and the accompanying ability to exploit redundancies in the signal.

### 5.5.1 Audio Signals

Application of pitch-synchronous Fourier and wavelet approaches to single-voice audio has been discussed throughout this chapter. These models provide compact representations that enable a wide range of modifications. In polyphonic audio, pitch-based methods are not as immediately applicable since a repetitive time-domain structure may not exist in the signal. In those cases it would be necessary to first carry out *source separation* to derive single voice components with well-defined pitches; source separation is a difficult problem that has been addressed in both the signal processing community and in the psychoacoustics literature in considerations of *auditory scene analysis* [147, 207]. Given these difficulties, the PSWT is in essence primarily useful for the single voice case, which is relevant to speech coding and music synthesis; for instance, data compression can be achieved in samplers by using the signal estimate provided by the PSWT.

### 5.5.2 Electrocardiogram Signals

Electrocardiogram (ECG) signals, *i.e.* heartbeat signals, exhibit pseudo-periodic behavior. Nearby pulses are very similar in shape, but of course various evolutionary changes in the behavior are medically significant. It is important, then, to monitor the heartbeat signal and record it for future analysis, especially in ambulatory scenarios where a diagnostic expert may not be present. For such applications, as in all data storage scenarios, it is both economically and pragmatically important to store the data in a compressed format while preserving its salient features. Various methods of ambulatory ECG signal compression have been presented in the literature; these rely on either the redundancy between neighboring samplings of the signal or the redundancy between adjacent periods [208, 209]. Recently, a method exploiting both forms of redundancy was proposed [37]; here, the signal is segmented into pulses and arranged into a structure resembling a PSR matrix. Then, this structure is interpreted as an image and compressed using a two-dimensional discrete cosine transform (DCT); the compression is structured such that important features of the pulse shape are represented accurately. The pitch-synchronous approaches discussed in this chapter, especially the extension to two-dimensional wavelets, provide a similar approach; important features such as attack transients can be preserved in the representation. Both this DCT-based ECG compression algorithm and the PSWT itself are reminiscent of several other efforts involving multidimensional processing of one-dimensional signals, for instance image processing of audio [128, 210].

## 5.6 Conclusion

For pseudo-periodic signals, the redundancies between adjacent periods can be exploited to achieve compact signal models. This notion was the basic theme of this chapter, which opened with a discussion of estimation of signal periodicity and construction of a pitch-synchronous representation. This representation, which is itself useful for signal modification because of its granularity, primarily served to establish a framework for pitch-synchronous processing. Specifically, it was shown that using a pitch-synchronous representation in conjunction with sinusoidal modeling leads to a simpler analysis-synthesis and more compact models than in the general case. Furthermore, it was shown that the wavelet transform, which is intrinsically unsuitable for wideband harmonic signals, can be cast into a pitch-synchronous framework to yield effective models of pseudo-periodic signals. In either case, the model improvement is a result of the signal adaptivity brought about by extraction of the pitch parameter.

# Chapter 6

# Matching Pursuit and Atomic Models

In atomic models, a signal is represented in terms of localized time-frequency components. Chapter 3 discussed an interpretation of the sinusoidal model as an atomic decomposition in which the atoms are derived by extracting parameters from the signal; this perspective clarified the resolution tradeoffs in the model and motivated multiresolution extensions. In this chapter, signal-adaptive parametric models based on overcomplete dictionaries of time-frequency atoms are considered. Such overcomplete expansions can be derived using the matching pursuit algorithm [38]. The resulting representations are signal-adaptive in that the atoms for the model are chosen to match the signal behavior; furthermore, the models are parametric in that the atoms can be described in terms of simple parameters. The pursuit algorithm is reviewed in detail and variations are described; primarily, the method is formalized for the case of dictionaries of damped sinusoids, for which the computation can be carried out with simple recursive filter banks. Atoms based on damped sinusoids are shown to be more effective than symmetric Gabor atoms for representing transient signal behavior such as attacks in music.

## 6.1 Atomic Decompositions

Time-frequency atomic signal representations have been of growing interest since their introduction by Gabor several decades ago [71, 72]. The fundamental notions of atomic modeling are that a signal can be decomposed into elementary functions that are localized in time-frequency and that such decompositions are useful for applications such as signal analysis and coding. This section provides an overview of the computation and properties of atomic models. The overview is based on an interpretation of atomic modeling as a linear algebraic inverse problem, which is discussed below.

### 6.1.1 Signal Modeling as an Inverse Problem

As discussed in Chapter 1, a signal model of the form

$$x[n] \;=\; \sum_{m=1}^{M} \alpha_m d_m[n] \tag{6.1}$$

can be expressed in matrix notation as

$$x \;=\; D\,\alpha \quad \text{with} \quad D \;=\; [d_1\; d_2 \cdots d_m \cdots d_M]\,, \tag{6.2}$$

where the signal $x$ is a column vector ($N \times 1$), $\alpha$ is a column vector of expansion coefficients ($M \times 1$), and $D$ is an $N \times M$ matrix whose columns are the expansion functions $d_m[n]$. Derivation of the model coefficients thus corresponds to an inverse problem.

When the functions $\{d_m[n]\}$ constitute a basis, such as in Fourier and wavelet decompositions, the matrix $D$ in Equation (6.2) is square ($N = M$) and invertible and the expansion coefficients $\alpha$ for a signal $x$ are uniquely given by

$$\alpha \;=\; D^{-1}x. \tag{6.3}$$

In the framework of biorthogonal bases, there is a dual basis $\tilde{D}$ such that

$$D^{-1} \;=\; \tilde{D}^H \quad \text{and} \quad \alpha \;=\; \tilde{D}^H x. \tag{6.4}$$

For orthogonal bases, $\tilde{D} = D$. Considering one component in Equation (6.4), it is clear that the coefficients in a basis expansion can each be derived independently using the formula

$$\alpha_m \;=\; \tilde{d}_m^H\, x \;=\; \langle \tilde{d}_m, x\rangle. \tag{6.5}$$

While this ease of computation is an attractive feature, basis expansions are not generally useful for modeling arbitrary signals given the drawbacks demonstrated in Section 1.4.1; namely, basis expansions do not provide compact models of arbitrary signals. This shortcoming results from the attempt to model arbitrary signals in terms of a limited and fixed set of functions.

To overcome the difficulties of basis expansions, signals can instead be modeled using an overcomplete set of atoms that exhibits a wide range of time-frequency behaviors [38, 68, 42, 43, 211]. Such overcomplete expansions allow for compact representation of arbitrary signals for the sake of compression or analysis [38, 92]. With respect to the interpretation of signal modeling as an inverse problem, when the functions $\{d_m[n]\}$ constitute an overcomplete or redundant set ($M > N$), the dictionary matrix $D$ is of rank $N$ and the linear system in Equation (6.2) is underdetermined. The null space of $D$ then has nonzero dimension and there are an infinite number of expansions of the form of Equation (6.1). Various methods of deriving overcomplete expansions are discussed in the next section; specifically, it is established that sparse approximate solutions of an inverse problem correspond to compact signal models, and that computation of such sparse models calls for a nonlinear approach.

### 6.1.2 Computation of Overcomplete Expansions

As described in Section 1.4.2, there are a variety of frameworks for deriving overcomplete signal expansions; these differ in the structure of the dictionary and the manner in which dictionary atoms are selected for the expansion. Examples include best basis methods and adaptive wavelet packets, where the overcomplete dictionary consists of a collection of bases; a basis for a signal expansion is chosen from the set of bases according to a metric such as entropy or rate-distortion [40, 41, 60]. In this chapter, signal decomposition using more general overcomplete sets is considered. Such approaches can be roughly grouped into two categories: parallel methods such as the method of frames [63, 70], basis pursuit [42, 43], and FOCUSS [68, 67], in which computation of the various expansion components is coupled; and, sequential methods such as matching pursuit and its variations [38, 68, 211, 212, 213, 214], in which models are computed one component at a time. All of these methods can be interpreted as approaches to solving inverse problems. For compact signal modeling, sparse approximate solutions are of interest; the matching pursuit algorithm of [38] is particularly useful since it is amenable to task of modeling arbitrary signals using parameterized time-frequency atoms in a successive refinement framework. After a brief review of the singular value decomposition and the pseudo-inverse, nonlinear approaches such as matching pursuit are motivated.

#### The SVD and the pseudo-inverse

One solution to arbitrary inverse problems can be arrived at using the singular value decomposition of the dictionary matrix, from which the pseudo-inverse $D^+$ can be derived [58]. The coefficient vector $\bar{\alpha} = D^+x$ has the minimum two-norm of all solutions [58]. This minimization of the two-norm is inappropriate for deriving signal models, however, in that it tends to spread energy throughout all of the elements of $\bar{\alpha}$. Such spreading undermines the goal of compaction.

An example of the dispersion of the SVD approach was given earlier in Figure 1.5. Figure 6.1 shows an alternative example in which the signal in question is constructed as the sum of two functions from an overcomplete set, meaning that there is an expansion in that overcomplete set with only two nonzero coefficients. This exact sparse expansion is shown in the plot by the asterisks; the dispersed expansion computed using the SVD pseudo-inverse is indicated by the circles. The representations can be immediately compared with respect to two applications: first, the sparse model is clearly more appropriate for compression; second, it provides a more useful analysis of the signal in that it identifies fundamental signal structures. This simulation thus provides an example of an issue discussed in Chapter 1, namely that compression and analysis are linked.

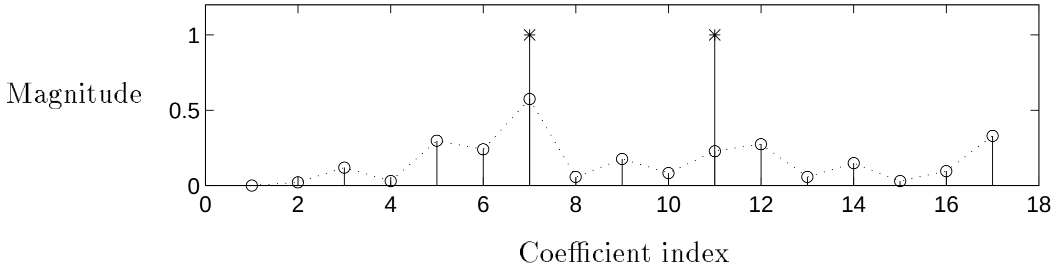

FIGURE 6.1: Overcomplete expansions and compaction. An exact sparse expansion of a signal in an overcomplete set ($*$) and the dispersed expansion given by the SVD pseudo-inverse ($\circ$).

### Sparse approximate solutions and compact models

Given the desire to derive compact representations for signal analysis, coding, denoising, and modeling in general, the SVD is not a particularly useful tool. An SVD-based expansion is by nature not sparse, and thresholding small expansion coefficients to improve the sparsity is not a useful approach [215, 69]. A more appropriate paradigm for deriving an overcomplete expansion is to apply an algorithm specifically designed to arrive at sparse solutions. Because of the complexity of the search, however, it is not computationally feasible to derive an optimal sparse expansion that perfectly models a signal. It is likewise not feasible to compute approximate sparse expansions that minimize the error for a given sparsity; this is an NP-hard problem [39]. For this reason, it is necessary to narrow the considerations to methods that either derive sparse approximate solutions according to suboptimal criteria or derive exact solutions that are not optimally sparse. The matching pursuit algorithm introduced in [38] is an example of the former category; it is the method of choice here since it provides a framework for deriving sparse approximate models with successive refinements and since it can be implemented with low cost as will be seen. Methods of the latter type tend to be computationally costly and to lack an effective successive refinement framework [42, 67].

## 6.1.3 Signal-Adaptive Parametric Models

The set of expansion coefficients and functions in Equation (6.1) provides a model of the signal. If the model is compact or sparse, the decomposition indicates fundamental signal features and is useful for analysis and coding. Such compact models necessarily involve expansion functions that are highly correlated with the signal; this property is an indication of signal adaptivity.

As discussed throughout this thesis, effective signal models can be achieved by using signal-adaptive expansion functions, *e.g.* the multiresolution sinusoidal partials of

Chapter 3 or the pitch-synchronous grains of Chapter 5. In those approaches, model parameters are extracted from the signal by the analysis process and the synthesis expansion functions are constructed using these parameters; in such methods, the parameter extraction leads to the signal adaptivity of the representation. In atomic models based on matching pursuit with an overcomplete dictionary, signal adaptivity is instead achieved by choosing expansion functions from the dictionary that match the time-frequency behavior of the signal. Using a highly overcomplete set of time-frequency atoms enables compact representation of a wide range of time-frequency behaviors. Furthermore, when the dictionary has a parametric structure, *i.e.* when the atoms in the dictionary can be indexed by meaningful parameters, the resultant model is both signal-adaptive and parametric. While this framework is fundamentally different from that of traditional parametric models, the signal models in the two cases have similar properties.

## 6.2 Matching Pursuit

Matching pursuit is a greedy iterative algorithm for deriving signal decompositions in terms of expansion functions chosen from a dictionary [38]. To achieve compact representation of arbitrary signals, it is necessary that the dictionary elements or atoms exhibit a wide range of time-frequency behaviors and that the appropriate atoms from the dictionary be chosen to decompose a particular signal. When a well-designed overcomplete dictionary is used in matching pursuit, the nonlinear nature of the algorithm leads to compact signal-adaptive models [38, 211, 92].

A dictionary can be likened to the matrix $D$ in Equation (6.2) by considering the atoms to be the matrix columns; then, matching pursuit can be interpreted as an approach for computing sparse approximate solutions to inverse problems [69, 215]. For an overcomplete dictionary, the linear system is underdetermined and an infinite number of solutions exist. As discussed in Section 6.1.2, sparse approximate solutions are useful for signal analysis, compression, and enhancement. Since such solutions are not provided by traditional linear methods such as the SVD, a nonlinear approximation paradigm such as matching pursuit is called for [38, 215, 69, 92].

### 6.2.1 One-Dimensional Pursuit

The greedy iteration in the matching pursuit algorithm is carried out as follows. First, the atom that best approximates the signal is chosen, where the two-norm is used as the approximation metric because of its mathematical convenience. The contribution of this atom is then subtracted from the signal and the process is iterated on the residual. Denoting the dictionary by $D$ since it corresponds to the matrix $D$ in Equation (6.2), the task at the $i$-th stage of the algorithm is to find the atom $d_{m(i)}[n] \in D$ that minimizes the two-norm of the residual signal

$$r_{i+1}[n] \;=\; r_i[n] - \alpha_i d_{m(i)}[n], \tag{6.6}$$

where $\alpha_i$ is a weight that describes the contribution of the atom to the signal, *i.e.* the expansion coefficient, and $m(i)$ is the dictionary index of the atom chosen at the $i$-th stage; the iteration begins with $r_1[n] = x[n]$. To simplify the notation, the atom chosen at the $i$-th stage is hereafter referred to as $g_i[n]$, where

$$g_i[n] \;=\; d_{m(i)}[n] \tag{6.7}$$

from Equation (6.6). The subscript $i$ refers to the iteration when $g_i[n]$ was chosen, while $m(i)$ is the actual dictionary index of $g_i[n]$.

Treating the signals as column vectors, the optimal atom to choose at the $i$-th stage can be expressed as

$$g_i = \arg\min_{g_i \in D} ||r_{i+1}||^2 = \arg\min_{g_i \in D} ||r_i - \alpha_i g_i||^2. \tag{6.8}$$

The orthogonality principle gives the value of $\alpha_i$:

$$\langle r_{i+1}, g_i \rangle \;=\; \langle r_i - \alpha_i g_i, g_i \rangle \;=\; (r_i - \alpha_i g_i)^H g_i \;=\; 0 \tag{6.9}$$

$$\Longrightarrow \;\alpha_i \;=\; \frac{\langle g_i, r_i \rangle}{\langle g_i, g_i \rangle} \;=\; \frac{\langle g_i, r_i \rangle}{||g_i||^2} \;=\; \langle g_i, r_i \rangle, \tag{6.10}$$

where the last step follows from restricting the atoms to be unit-norm. The norm of $r_{i+1}[n]$ can then be expressed as

$$||r_{i+1}||^2 \;=\; ||r_i||^2 \;-\; \frac{|\langle g_i, r_i \rangle|^2}{||g_i||^2} \;=\; ||r_i||^2 \;-\; |\alpha_i|^2, \tag{6.11}$$

which is minimized by maximizing the metric

$$\Psi \;=\; |\alpha_i|^2 \;=\; |\langle g_i, r_i \rangle|^2, \tag{6.12}$$

which is equivalent to choosing the atom whose inner product with the signal has the largest magnitude; Equation (6.8) can thus be rewritten as

$$g_i = \arg\max_{g_i \in D} \Psi = \arg\max_{g_i \in D} |\langle g_i, r_i \rangle|. \tag{6.13}$$

An example of this optimization is illustrated in Figure 6.2. Note that Equation (6.11) shows that the norm of the residual decreases as the algorithm progresses provided that an exact model has not been reached and that the dictionary is complete; for an undercomplete dictionary, the residual may belong to a subspace that is orthogonal to all of the dictionary vectors, in which case the model cannot be further improved by pursuit.

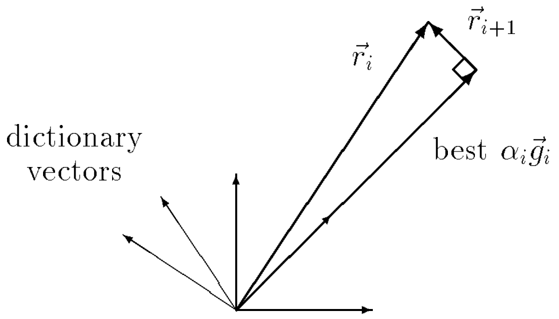

FIGURE 6.2: Matching pursuit and the orthogonality principle. The two-norm or Euclidean length of $r_{i+1}$ is minimized by choosing $g_i$ to maximize the metric $|\langle g_i, r_i\rangle|$ and $\alpha_i$ such that $\langle r_{i+1}, g_i\rangle = 0$.

In deriving a signal decomposition, the matching pursuit is iterated until the residual energy is below some threshold or until some other halting criterion is met. After $I$ iterations, the pursuit provides the sparse approximate model

$$x[n] \;\approx\; \sum_{i=1}^{I} \alpha_i g_i[n] \;=\; \sum_{i=1}^{I} \alpha_i d_{m(i)}[n]. \tag{6.14}$$

As indicated in Equation (6.11), the mean-squared error of the model decreases as the number of iterations increases [38]. This convergence implies that $I$ iterations will yield a reasonable $I$-term model; this model, however, is in general not optimal in the mean-squared sense because of the term-by-term greediness of the algorithm. Computing the optimal $I$-term estimate using an overcomplete dictionary requires finding the minimum projection error over all $I$-dimensional dictionary subspaces, which is an NP-hard problem as mentioned earlier; this complexity result is established in [39] by relating the optimal approximation problem to the exact cover by 3-sets problem, which is known to be NP-complete.

To enable representation of a wide range of signal features, a large dictionary of time-frequency atoms is used in the matching pursuit algorithm. The computation of the correlations $\langle g, r_i\rangle$ for all $g \in D$ is thus costly. As noted in [38], this computation can be substantially reduced using an update formula based on Equation (6.6); the correlations at stage $i+1$ are given by

$$\langle g, r_{i+1}\rangle \;=\; \langle g, r_i\rangle \;-\; \alpha_i\langle g, g_i\rangle, \tag{6.15}$$

where the only new computation required for the correlation update is the dictionary cross-correlation term $\langle g, g_i\rangle$, which can be precomputed and stored if enough memory is available. This is discussed further in Section 6.4.3.

It should be noted that matching pursuit is similar to some forms of vector quantization [216] and is related to the projection pursuit method investigated earlier in the field of statistics for the task of finding compact models for data sets [217, 218]. Furthermore, such greedy approximation methods have been considered in linear algebra applications for some time [69, 219].

### 6.2.2 Subspace Pursuit

Though searching for the optimal high-dimensional subspace is not reasonable, it is worthwhile to consider the related problem of finding an optimal low-dimension subspace at each iteration of the pursuit, especially if the subspaces under consideration exhibit a simplifying structure. In this variation of the algorithm, the $i$-th iteration consists of searching for an $N \times R$ matrix $G$, whose $R$ columns are dictionary atoms, that minimizes the two-norm of the residual $r_{i+1} = r_i - G\alpha$, where $\alpha$ is an $R \times 1$ vector of weights. This $R$-dimensional formulation is similar to the one-dimensional case; the orthogonality constraint $\langle r_i - G\alpha, G \rangle = 0$ leads to a solution for the weights:

$$\alpha = \left(G^H G\right)^{-1} G^H r_i. \tag{6.16}$$

The energy of the residual is then given by

$$\langle r_{i+1}, r_{i+1} \rangle = \langle r_i, r_i \rangle - r_i^H G \left(G^H G\right)^{-1} G^H r_i, \tag{6.17}$$

which is minimized by choosing $G$ so as to maximize the second term. This approach is expensive unless $G$ consists of orthogonal vectors or has some other special structure.

### 6.2.3 Conjugate Subspaces

One useful subspace to consider in subspace pursuit is the two-dimensional subspace spanned by an atom and its complex conjugate. Here, the two columns of $G$ are simply an atom $g$ and its conjugate $g^*$. Assuming that the signal $r_i$ is real and that $g$ has nonzero real and imaginary parts so that $G$ has full column rank and $G^H G$ is invertible, the results given in the previous section can be significantly simplified. Letting $, = \langle g, g^* \rangle$ and $\beta = \langle g, r_i \rangle$, the metric to maximize through the choice of $g$ is

$$\Psi = \frac{1}{1 - |,|^2}\left(2|\beta|^2 - ,(\beta^*)^2 - ,^*\beta^2\right) \tag{6.18}$$

and the optimal weights are

$$\alpha = \begin{bmatrix} \alpha(1) \\ \alpha(2) \end{bmatrix} = \frac{1}{1 - |,|^2} \begin{bmatrix} \beta - ,\beta^* \\ \beta^* - ,^*\beta \end{bmatrix}. \tag{6.19}$$

Note that the above metric can also be written as

$$\Psi = \beta^*\alpha(1) + \beta\alpha(1)^* \tag{6.20}$$

and that $\alpha(1) = \alpha(2)^*$, meaning that the algorithm simply searches for the atom $g_i$ that minimizes the two-norm of the residual

$$\begin{aligned} r_{i+1}[n] &= r_i[n] - \alpha_i(1) g_i[n] - \alpha_i(1)^* g_i^*[n] \qquad (6.21) \\ &= r_i[n] - 2\Re\{\alpha_i(1) g_i[n]\}, \qquad (6.22) \end{aligned}$$

which is real-valued; the orthogonal projection of a real signal onto the subspace spanned by a conjugate pair is again real.

The decompositions that result from considering conjugate subspaces are of the form

$$x \approx 2\sum_{i=1}^{I} \Re\left\{\alpha_i(1) g_i[n]\right\}. \qquad (6.23)$$

This approach provides real decompositions of real signals using an underlying complex dictionary. The same notion is discussed briefly in [38] based on a different computational framework.

For dictionaries consisting of both complex and purely real (or imaginary) atoms, the real atoms must be considered independently of the various conjugate subspaces since the above formulation breaks down when $g$ and $g^*$ are linearly dependent; in that case, $|,| = 1$ and the matrix $G$ is singular. It is thus necessary to compare metrics of the form given in Equations (6.18) and (6.20) for conjugate subspaces with metrics of the form $|\langle g, r_i\rangle|^2$ from Equation (6.12) for real atoms. These metrics quantify the amount of energy removed from the residual in either case, and thus provide for a fair choice between conjugate subspaces and real atoms in the pursuit decomposition.

### 6.2.4 Orthogonal Matching Pursuit

As depicted in Figure 6.2, the matching pursuit algorithm relies on the orthogonality principle. At stage $i$, the residual $r_i$ is projected onto the atom $g_i$ such that the new residual $r_{i+1}$ is orthogonal to $g_i$. If the dictionary is highly overcomplete and its elements populate the signal space densely, the first few atoms chosen for a decomposition tend to be orthogonal to each other, meaning that successive projection operations extract independent signal components. Later iterations, however, do not exhibit this tendency; the selected atoms are no longer orthogonal to previously chosen atoms and the projection actually reintroduces components extracted by the early atoms. This problem of *readmission* is addressed in orthogonal matching pursuit and its variations; the fundamental idea is to explicitly orthogonalize the functions chosen for the expansion.

#### Backward orthogonal matching pursuit

Orthogonal matching pursuit is a basic variation of the matching pursuit algorithm [212]. In this method, the $i$-th stage is initiated by selecting an atom $g_i$ according to the correlation metric as in the standard pursuit; then, rather than orthogonalizing the residual $r_{i+1}$ with respect to the single atom $g_i$, the residual is orthogonalized with respect to the subspace spanned by the atoms chosen for the expansion up to and including the $i$-th stage, *i.e.* the atoms $g_1, g_2, \ldots, g_i$. To achieve this orthogonalization, however, it is necessary to modify all of the expansion coefficients at each stage. This issue is clarified by interpreting orthogonal matching pursuit as a subspace pursuit in which the space is iteratively grown. In terms of the discussion of subspace pursuit in Section 6.2.2, the subspace matrix is

$$G_i \;=\; [g_1\, g_2\, \cdots\, g_i] \tag{6.24}$$

and the orthogonalization criterion is

$$\langle r_{i+1}, G_i \rangle \;=\; \langle x - G_i\alpha_i, G_i \rangle \;=\; 0. \tag{6.25}$$

This constraint can be used to derive the appropriate vector of coefficients $\alpha_i$ for the subspace projection, namely

$$\alpha_i \;=\; \left(G_i^H G_i\right)^{-1} G_i^H x, \tag{6.26}$$

which differs from Equation (6.16) in that the coefficients are derived as a function of the original signal $x$ and not as a function of the residual $r_i$. The correlation metric for atom selection, however, is based on the residual signal as in one-dimensional pursuit. Note that the inverse $\left(G_i^H G_i\right)^{-1}$ can be computed recursively using the matrix inversion lemma and the inverse $\left(G_{i-1}^H G_{i-1}\right)^{-1}$ computed at the previous stage [212].

At any given stage of an orthogonal pursuit, derivation of the new set of expansion coefficients can be interpreted as a Gram-Schmidt orthogonalization carried out on the new atom chosen for the expansion. This interpretation can be established in an inductive manner by first assuming that the atoms at stage $i-1$ have been orthogonalized by a Gram-Schmidt process; in other words, assume that the matrix $G_{i-1}$ has been converted into a matrix $\bar{G}_{i-1}$ with the same column space but with orthogonal columns. In this framework, the signal approximation at stage $i-1$ can be expressed as

$$x \;\approx\; \bar{G}_{i-1}\alpha_{i-1}, \tag{6.27}$$

where

$$\alpha_{i-1} \;=\; \bar{G}_{i-1}^H x \tag{6.28}$$

since the columns of $\bar{G}_{i-1}$ are orthogonal; note that $\bar{G}_{i-1}$ is an $N \times i-1$ matrix. At stage $i$, a new unit-norm atom $g_i$ is chosen for the expansion according to the magnitude correlation metric of Equation (6.12); then, the approximation error is minimized by projecting the signal onto the subspace spanned by the columns of the new matrix

$$G_i \;=\; [\bar{G}_{i-1} \;\; g_i]. \tag{6.29}$$

Using the solution for the coefficients given in Equation (6.26), the $i$-term signal decomposition can be written as

$$G_i\alpha_i \;=\; [\bar{G}_{i-1}\;\; g_i]\,\alpha_i \;=\; [\bar{G}_{i-1}\;\; g_i]\left(G_i^H G_i\right)^{-1} G_i^H x \tag{6.30}$$

$$= [\bar{G}_{i-1}\;\; g_i]\begin{bmatrix} I_{i-1} & \bar{G}_{i-1}^H g_i \\ g_i^H \bar{G}_{i-1} & 1 \end{bmatrix}^{-1}\begin{bmatrix} \alpha_{i-1} \\ g_i^H x \end{bmatrix} \tag{6.31}$$

$$= [\bar{G}_{i-1}\;\; g_i]\begin{bmatrix} I_{i-1} + \frac{\Gamma\Gamma^H}{1-\Gamma^H\Gamma} & \frac{-\Gamma}{1-\Gamma^H\Gamma} \\ \frac{-\Gamma^H}{1-\Gamma^H\Gamma} & \frac{1}{1-\Gamma^H\Gamma} \end{bmatrix}\begin{bmatrix} \alpha_{i-1} \\ g_i^H x \end{bmatrix}, \tag{6.32}$$

where , $= \bar{G}_{i-1}^H g_i$ and $I_n$ denotes an identity matrix of size $n \times n$. The decomposition can then be simplified to

$$G_i\alpha_i \;=\; \bar{G}_{i-1}\alpha_{i-1} \;+\; \frac{(\bar{G}_{i-1}\bar{G}_{i-1}^H - I_N)g_i g_i^H(\bar{G}_{i-1}\bar{G}_{i-1}^H - I_N)x}{1 - g_i^H \bar{G}_{i-1}\bar{G}_{i-1}^H g_i} \tag{6.33}$$

$$= \bar{G}_{i-1}\alpha_{i-1} \;+\; \bar{g}_i\bar{g}_i^H x \;=\; \sum_{j=1}^{i}\langle \bar{g}_j, x\rangle\, \bar{g}_j, \tag{6.34}$$

where

$$\bar{g}_i \;=\; \frac{(\bar{G}_{i-1}\bar{G}_{i-1}^H - I_N)g_i}{\sqrt{1 - g_i^H \bar{G}_{i-1}\bar{G}_{i-1}^H g_i}}. \tag{6.35}$$

The vector $\bar{g}_i$ has unit norm and is orthogonal to the columns of the matrix $\bar{G}_{i-1}$. This orthogonality, combined with the initial orthogonality of the columns of $\bar{G}_{i-1}$, indicates that the final expression in Equation (6.34) is a basis expansion in an $i$-dimensional subspace of the signal space. It is not a basis expansion of the original signal; it is an approximation of the signal in the subspace spanned by $g_1, g_2, \ldots, g_i$, for which an orthogonal basis has been derived by the Gram-Schmidt method. For $i = N$, this subspace is equivalent to the signal space and a perfect representation is achieved.

The Gram-Schmidt orthogonalization discussed above need not be explicit in an implementation of orthogonal matching pursuit. As mentioned earlier, the pursuit can be carried out with reference to the original dictionary atoms by updating the inverse using the matrix inversion lemma; this approach preserves the parametric nature of the expansion, which would be compromised if the atoms were explicitly modified via the Gram-Schmidt process. Furthermore, note that this algorithm corrects for readmitted components in the orthogonalization step. Since this corrective orthogonalization is carried out after the atom selection, the algorithm can be referred to as a *backward* method; this designation serves to differentiate it from the *forward* approach discussed in the next section.

A number of variations of orthogonal matching pursuit can be envisioned. For instance, the orthogonalization need not be carried out every iteration. In the limit, an expansion given by a one-dimensional pursuit can be orthogonalized after its last iteration by projecting the original signal onto the subspace spanned by the iteratively chosen expansion functions. This projection operation minimizes the error of the residual for approximating the signal using those particular expansion functions; this approximation is however not necessarily a globally optimal sparse model.

In the literature, speech coding using orthogonal matching pursuit has been discussed [220]. Furthermore, a number of refinements of the algorithm have been proposed and explored [68, 214]. Such refinements basically involve different ways in which orthogonality is imposed or exploited; for instance, orthogonal components can be evaluated simultaneously as in basis expansions [214]. The following section discusses a method which employs the Gram-Schmidt procedure in a different way than the backward pursuit described above.

### Forward orthogonal matching pursuit

In orthogonal matching pursuit as proposed in [212], which corresponds to the backward pursuit described above, the atom $g_i$ is chosen irrespective of the subspace spanned by the first $i-1$ atoms, *i.e.* the column space of $G_{i-1}$, and then orthogonalization is carried out. Given a decomposition with $i-1$ atoms, however, the approximation error of the succeeding $i$-term model can be decreased if the choice of atom is conditioned on $G_{i-1}$. As will be seen, this conditioning leads to a forward orthogonalization of the dictionary; in other words, the dictionary is orthogonalized prior to atom selection.

Using a similar induction framework as above, where $\bar{G}_{i-1}$ is assumed to have orthogonal columns, the $i$-term expansion can be expressed as in Equation (6.33); the difference in the forward algorithm is that the atom $g_i$ has not yet been selected. Rather than choosing the atom to maximize the magnitude of the correlation $\langle g, r_i \rangle$ as above, in this approach the atom is chosen to maximize the metric

$$\psi \;=\; x^H G_i \left(G_i^H G_i\right)^{-1} G_i^H x, \tag{6.36}$$

which corresponds to the second term in Equation (6.17) from the general development of subspace pursuit. For this specific case, where the subspace is again iteratively grown, the metric can be expressed as

$$\begin{aligned} \psi \;&=\; x^H \left[ \bar{G}_{i-1}\bar{G}_{i-1}^H \;+\; \frac{(\bar{G}_{i-1}\bar{G}_{i-1}^H - I_N) g_i g_i^H (\bar{G}_{i-1}\bar{G}_{i-1}^H - I_N)}{1 - g_i^H \bar{G}_{i-1}\bar{G}_{i-1}^H g_i} \right] x & (6.37) \\ &=\; x^H \left[ \bar{G}_{i-1}\bar{G}_{i-1}^H \; + \bar{g}_i \bar{g}_i^H \right] x, & (6.38) \end{aligned}$$

where $\bar{g}_i$ is as given in Equation (6.35). In the earlier formulation, $g_i$ was chosen and $\bar{g}_i$ was derived from that choice so as to be orthogonal to the columns of $\bar{G}_{i-1}$. In this case, on the other hand, all possible $\bar{g}_i$ are considered for the expansion, and the one which maximizes the metric $\Psi$ is chosen. Given some $\bar{G}_{i-1}$, the $i$-term approximation error resulting from this choice of $\bar{g}_i$ will always be less than or equal to the error of the $i$-term approximation arrived at in the backward orthogonal matching pursuit.

Note that all of the atoms $\bar{g}_i$ are orthogonal to $\bar{G}_{i-1}$ by construction. This observation suggests an interpretation of this variation of orthogonal matching pursuit. Specifically, this approach is equivalent to carrying out a Gram-Schmidt orthogonalization on the dictionary at each stage. Once an atom is chosen from the dictionary for the expansion, the dictionary is orthogonalized with respect to that atom; in the next stage, correlations with the orthogonalized dictionary, namely $\langle \bar{g}, x \rangle$, are computed to find the atom that maximizes the metric $\Psi$. This orthogonalization process completely prevents readmission, but at the cost of added computation to maintain the changing dictionary.

### Greedy algorithms and computation–rate–distortion

In matching pursuit and its orthogonal variations, each iteration attempts to improve the signal approximation as much as possible by minimizing some error metric. In orthogonal pursuits, the metric depends on previous iterations; in any case, however, the approximation is made without regard to future iterations. Matching pursuit is thus categorized as a *greedy* algorithm. It is well known that such greedy algorithms, when applied to overcomplete dictionaries, do not lead to optimal approximations, *i.e.* optimal compact models; however, greedy approaches are justified given the complexity of optimal approximation [39, 69]. Furthermore, it should be noted as in Section 6.1.2 that the use of a greedy algorithm inherently leads to successive refinement, which is a desirable property in signal models.

For the application of compact signal modeling, it is of interest to compare the approximation errors of matching pursuit and the backward and forward orthogonal pursuits. This comparison, however, can only be made in a definitive sense for the case where each algorithm is initiated at stage $i$ with the same first $i-1$ atoms; then, the energy removed from the residual in the forward case is always greater than or equal to that in the backward approach, which is in turn greater than or equal to that in the standard pursuit. Conditioned on the first $i-1$ terms, the forward approach provides the optimal $i$-term approximation. For the case of arbitrary $i$-term decompositions, however, no absolute comparison can be made between the algorithms. While error bounds can be established for the various greedy approximations, the relative performance for a given signal cannot be guaranteed *a priori* since the algorithms use different strategies for selecting atoms [221, 222, 223]. Useful predictive comparisons of the algorithms can be carried out using ensemble results based on random dictionaries [68].

In the preceding paragraph, as in most discussions of signal modeling, comparisons between models are phrased in terms of the amount of information required to describe a certain model, *i.e.* the compaction, and the approximation error of the model. This rate-distortion tradeoff is the typical metric by which models are compared. In implementations, however, it is also important to account for the resources required for model computation. In general, a model can achieve a better rate-distortion characteristic through increased computation. For example, recall from the earlier discussion that an approximation provided by a standard pursuit can be improved after the last stage by backward orthogonalization of the full expansion; this process results in a lower distortion at a fixed rate at the expense of the computation of the subspace projection. Given the preceding observation about the impact of computation and this supporting example, it is reasonable to assert that computation considerations are important in model comparisons. Examples of computation-distortion tradeoffs are given for the case of orthogonal matching pursuit in [212]; a preliminary treatment of general computation-rate-distortion theory is given in [224].

## 6.3 Time-Frequency Dictionaries

Matching pursuit yields a sparse approximate signal decomposition based on a dictionary of expansion functions. In a compact model, the atoms in the expansion necessarily correspond to basic signal features. This is especially useful for analysis and coding if the atoms can be described by meaningful parameters such as time location, frequency modulation, and scale; then, the basic signal features can be identified and parameterized. In this light, parametric overcomplete dictionaries consisting of atoms that exhibit a wide range of localized time and frequency behaviors are of great interest; matching pursuit then provides a compact, adaptive, and parametric time-frequency representation of a signal [38]. Such localized time-frequency atoms were introduced by Gabor from a theoretical standpoint and according to psychoacoustic motivations [71, 72].

### 6.3.1 Gabor Atoms

The literature on matching pursuit has focused on applications involving dictionaries of Gabor atoms since these are appropriate expansion functions for general time-frequency signal models [38]. [1] In continuous time, such atoms are derived from a single unit-norm window function $g(t)$ by scaling, modulation, and translation:

$$g_{\{s,\omega,\tau\}}(t) \;=\; \frac{1}{\sqrt{s}}g\left(\frac{n-\tau}{s}\right)e^{j\omega(n-\tau)}. \tag{6.39}$$

This definition can be extended to discrete time by a sampling argument as in [38]; fundamentally, the extension simply indicates that Gabor atoms can be represented in discrete

[1] Atoms corresponding to wavelet and cosine packets have also been considered [42, 212].

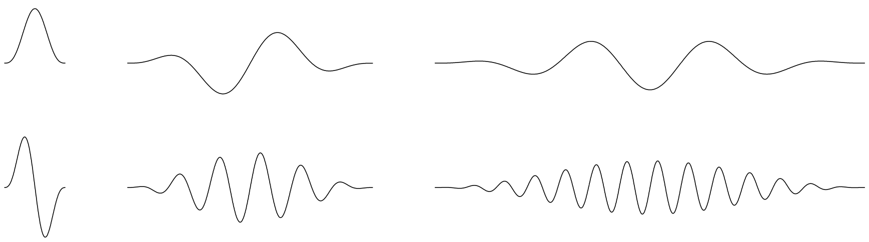

FIGURE 6.3: Symmetric Gabor atoms. Such time-frequency dictionary elements are derived from a symmetric window by scaling, modulation, and translation operations as described in Equation (6.39).

time as

$$g_{\{s,\omega,\tau\}}[n] \;=\; f_s[n-\tau_0]e^{j\omega(n-\tau)}, \tag{6.40}$$

where $f_s[n]$ is a unit-norm window function supported on a scale $s$. Examples are depicted in Figure 6.3.

Note that Gabor atoms are scaled to have unit-norm and that each is indexed in the dictionary by a parameter set $\{s,\omega,\tau\}$. This parametric structure allows for a simple description of a specific dictionary, which is useful for compression. When the atomic parameters are not tightly restricted, Gabor dictionaries are highly overcomplete and can include both Fourier and wavelet-like bases. One issue to note is that the modulation of an atom can be defined independently of the time shift, or *dereferenced*, as it will be referred to hereafter:

$$\tilde{g}_{\{s,\omega,\tau\}}[n] \;=\; \frac{1}{\sqrt{s}}g\left[\frac{n-\tau}{s}\right]e^{j\omega n} \;=\; e^{j\omega\tau}g_{\{s,\omega,\tau\}}[n]. \tag{6.41}$$

This simple phase relationship will have an impact in later considerations; note that this distinction between models of time is analogous to the issue discussed in Section 2.2.1 in the context of the STFT time reference.

In applications of Gabor functions, $g[n]$ is typically an even-symmetric window. The associated dictionaries thus consist of atoms that exhibit symmetric time-domain behavior. This is problematic for modeling asymmetric features such as transients, which occur frequently in natural signals such as music. Figure 6.4(a) shows a typical transient from linear system theory, the damped sinusoid; the first stage of a matching pursuit based on symmetric Gabor functions chooses the atom shown in Figure 6.4(b). This atom matches the frequency behavior of the signal, but its time-domain symmetry results in a pre-echo as indicated. The atomic model has energy before the onset of the original signal; as a result, the residual has both a pre-echo and a discontinuity at the onset time as shown in Figure 6.4(c). In later stages, then, the matching pursuit must incorporate small-scale atoms into the decomposition to remove the pre-echo and to model the discontinuity. One approach to this problem is the high-resolution matching pursuit algorithm

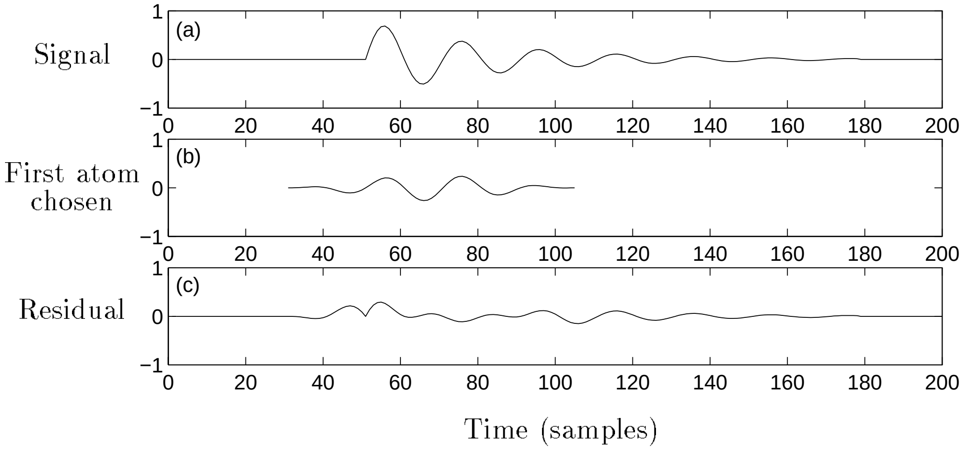

FIGURE 6.4: A pre-echo is introduced in atomics models of transient signals if the atoms are symmetric. The plots show (a) a damped sinusoidal signal, (b) the first atom chosen from a symmetric Gabor dictionary by matching pursuit, and (c) the residual. Note the pre-echo in the atomic model and the artifact in the residual at the onset time.

proposed in [213, 225], where symmetric atoms are generally still used but the selection metric is modified so that atoms that introduce drastic artifacts are not chosen for the decomposition. Fundamentally, however, symmetric functions are simply not well-suited for modeling asymmetric events. With that in mind, an alternative approach to modeling signals with transient behavior is to use a dictionary of asymmetric atoms, *e.g.* damped sinusoids. Such atoms are physically sensible given the common occurrence of damped oscillations in natural signals.

### 6.3.2 Damped Sinusoids

The common occurrence of damped oscillations in natural signals justifies considering damped sinusoids as building blocks in signal decompositions. The application at hand is further motivated in that damped sinusoids are better suited than symmetric Gabor atoms for modeling transients. Like the atoms in a general Gabor dictionary, damped sinusoidal atoms can be indexed by characteristic parameters, namely the damping factor $a$, modulation frequency $\omega$, and start time $\tau$:

$$g_{\{a,\omega,\tau\}}[n] \;=\; S_a\; a^{(n-\tau)} e^{j\omega(n-\tau)} u[n-\tau], \tag{6.42}$$

or, if the modulation is dereferenced,

$$\tilde{g}_{\{a,\omega,\tau\}}[n] \;=\; S_a\; a^{(n-\tau)} e^{j\omega n} u[n-\tau], \tag{6.43}$$

where the factor $S_a$ is included for unit-norm scaling. Examples are depicted in Figure 6.5. It should be noted that these atoms can be interpreted as Gabor functions derived from

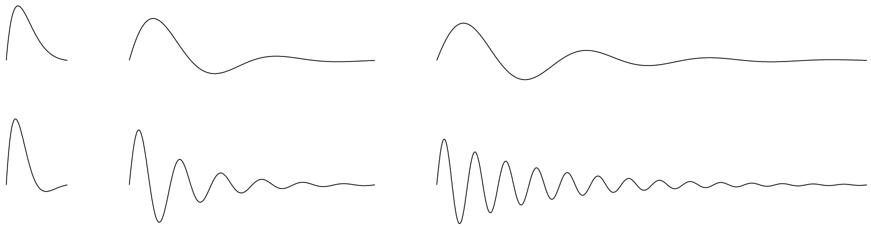

FIGURE 6.5: Damped sinusoids: Gabor atoms based on a one-sided exponential window.

a one-sided exponential window; they are just differentiated from typical Gabor atoms by their asymmetry. Also, the atomic structure is more readily indicated by a damping factor than a scale parameter, so the dictionary index set $\{a, \omega, \tau\}$ is used instead of the general Gabor set $\{s, \omega, \tau\}$.

For the sake of realizability, a damped sinusoidal atom is truncated when its amplitude falls below a threshold $T$; the corresponding length is $L = \lceil \log T / \log a \rceil$, and the appropriate scaling factor is then $S_a = \sqrt{(1-a^2)/(1-a^{2L})}$. Note that this truncation results in sensible localization properties; heavily damped atoms are short-lived, and lightly damped atoms persist in time. Also note that the atoms are one-sided; an atom corresponds to the impulse response of a filter with a single complex pole; this is a suitable property given the intent of representing transient signals, assuming that the source of the signal can be well-modeled by simple linear systems.

Several approaches in the literature have dealt with time-frequency atoms having exponential behavior. In [226], damped sinusoids are used to provide a time-frequency representation in which transients are identifiable. In the application outlined in [226], some prior knowledge of the damping factor is assumed, which is reasonable for detection applications but inappropriate for deriving decompositions of arbitrary signals; extensions of the algorithm, however, may prove useful for signal modeling. In [227], wavelets based on recursive filter banks are derived; these provide orthogonal expansions with respect to basis functions having infinite time support. This treatment focuses on the more general scenario of overcomplete expansions; unlike in the basis case, the constituent atoms have a flexible parametric structure.

### 6.3.3 Composite Atoms

The simple example of Figure 6.4 shows that symmetric atoms are inappropriate for modeling some signals. While the Figure 6.4 example is motivated by physical considerations, *i.e.* simple linear models of physical systems, it certainly does not encompass the wide range of complicated behaviors observed in natural signals. It is of course trivial to construct examples for which asymmetric atoms would prove similarly ineffective. Thus, given the task of modeling arbitrary signals (that might have been generated by complicated nonlinear systems), it can be argued that a wide range of both symmetric and asymmetric atoms should be present in the dictionary. Such *composite* dictionaries are considered here.

One approach to generating a composite dictionary is to simply merge a dictionary of symmetric atoms with a dictionary of damped sinusoids. The pursuit described in Section 6.2 can be carried out using such a dictionary. One caveat to note is that the atomic index set requires an additional parameter to specify which type of atom the set refers to. Furthermore, the nonuniformity of the dictionary introduces some difficulties in the computation and storage of the dictionary cross-correlations needed for the correlation update of Equation (6.15). Such computation issues will be discussed in Section 6.4.3; it is indicated there that the uniformity of the dictionary is coupled to the cost of the pursuit.

It is shown in Section 6.4.1 that correlations with damped sinusoidal atoms can be computed with low cost without using the update formula of Equation (6.15). The approach applies both to causal and anticausal damped sinusoids, which motivates considering two-sided atoms constructed by coupling causal and anticausal components. This construction can be used to generate symmetric and asymmetric atoms; furthermore, these atoms can be smoothed by simple convolution operations. Such atoms take the form

$$g_{\{a,b,J,\omega,\tau\}}[n] \;=\; f_{\{a,b,J\}}[n-\tau]e^{j\omega(n-\tau)}, \tag{6.44}$$

or, if the modulation is dereferenced,

$$\tilde{g}_{\{a,b,J,\omega,\tau\}}[n] \;=\; f_{\{a,b,J\}}[n-\tau]e^{j\omega n}, \tag{6.45}$$

where the amplitude envelope is a unit-norm function constructed using a causal and an anticausal exponential according to the formula

$$f_{\{a,b,J\}}[n] = S_{\{a,b,J\}}\left(a^n u[n] + b^{-n}u[-n] - \delta[n]\right) * h_J[n], \tag{6.46}$$

where $\delta[n]$ is subtracted because the causal and anticausal components, as written, overlap at $n = 0$. The function $h_J[n]$ is a smoothing window of length $J$; later considerations will be limited to the case of a rectangular window. A variety of composite atoms are depicted in Figure 6.6.

The unit-norm scaling factor $S_{\{a,b,J\}}$ for a composite atom is given by

$$S_{\{a,b,J\}} \;=\; \frac{1}{\sqrt{\Upsilon(a,b,J)}}, \tag{6.47}$$

where $\Upsilon(a,b,J)$ denotes the squared-norm of the atom prior to scaling:

$$\Upsilon(a,b,J) \;=\; \sum_n \left|\left(a^n u[n] + b^{-n}u[-n] - \delta[n]\right) * h_J[n]\right|^2, \tag{6.48}$$

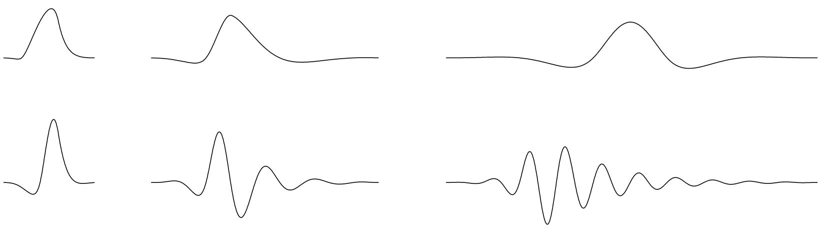

Figure 6.6: Composite atoms: Symmetric and asymmetric atoms constructed by coupling causal and anticausal damped sinusoids and using low-order smoothing.

which can be simplified to

$$\Upsilon(a,b,J) \;=\; \sum_{l=0}^{J-1}\sum_{k=0}^{J-1} \frac{a^{|l-k|}}{1-a^2} \;+\; \frac{b^{|l-k|}}{1-b^2} \;+\; \frac{a^{|l-k|}b - ab^{|l-k|}}{a-b}, \qquad (6.49)$$

which does not take truncation of the atoms into account. This approximation does not introduce significant errors if a small truncation threshold is used; furthermore, it should be noted that if some error is introduced, the iterative analysis-by-synthesis structure of matching pursuit corrects the error at a later stage. For the case of symmetric atoms $(a = b)$, the squared-norm can be written in closed form as

$$\Upsilon(a,a,J) \;=\; \frac{\left[J(1-a^4) + 2aJ(1-a^2)(1+a^J) - 4a(a^2+a+1)(1-a^J)\right]}{(1+a)(1-a)^3}, \qquad (6.50)$$

where a rectangular smoothing window has been assumed in the derivation. This scale factor affects the computational cost of the algorithm, but primarily with respect to precomputation. This issue will be examined in Sections 6.4.2 and 6.4.3.

The composite atoms described above can be written in terms of unit-norm constituent atoms:

$$\tilde{g}_{\{a,b,J,\omega,\tau\}}[n] \;=\; S_{\{a,b,J\}}\left(\frac{\tilde{g}^{+}_{\{a,\omega,\tau\}}[n]}{S_a} + \frac{\tilde{g}^{-}_{\{b,\omega,\tau\}}[n]}{S_b} - \delta[n]\right) * h_J[n] \qquad (6.51)$$

$$\;=\; \sum_{\Delta=0}^{J-1} \frac{\tilde{g}^{+}_{\{a,\omega,\tau+\Delta\}}[n]}{S_a} + \frac{\tilde{g}^{-}_{\{b,\omega,\tau+\Delta\}}[n]}{S_b} - \delta[n+\Delta], \qquad (6.52)$$

where $\tilde{g}^{+}_{\{a,\omega,\tau\}}[n]$ is a causal atom and $\tilde{g}^{-}_{\{b,\omega,\tau\}}[n]$ is an anticausal atom defined as

$$\tilde{g}^{-}_{\{b,\omega,\tau\}}[n] \;=\; S_b\, b^{-(n-\tau)} e^{j\omega n} u[-(n-\tau)]. \qquad (6.53)$$

Note that atoms with dereferenced modulation are used in the construction of Equation (6.52) so that the modulations add coherently in the sum over the time lags $\Delta$; in the other case, the constituent atoms must be summed with phase shifts $e^{j\omega\Delta}$ to achieve coherent modulation of the composite atom. As will be seen in Section 6.4.2, this atomic construction leads to a simple relationship between the correlations of the signal with the composite atom and with the underlying damped sinusoids, especially in the dereferenced case. Also, Equation (6.52) indicates the interplay of the various scale factors. Both of these issues will prove important for the computation considerations of Section 6.4.3.

The special case of symmetric atoms ($a = b$), one example of which is shown in Figure 6.6, suggests the use of this approach to construct atoms similar to symmetric Gabor atoms based on common windows. Given a unit-norm window function $w[n]$, the issue is to choose a damping factor $a$ and a smoothing order $J$ such that the resultant $f_{\{a,a,J\}}[n]$ accurately mimics $w[n]$. Using the two-norm as an accuracy metric, the objective is to minimize the error

$$\epsilon(a, J) \;=\; ||f_{\{a,a,J\}}[n] - w[n]||^2 \tag{6.54}$$

by optimizing $a$ and $J$. Since $f_{\{a,a,J\}}[n]$ and $w[n]$ are both unit-norm, this expression can be simplified to:

$$\epsilon(a, J) \;=\; 2\left(1 - \sum_n f_{\{a,a,J\}}[n]w[n]\right). \tag{6.55}$$

Not surprisingly, the overall objective of the optimization is thus to maximize the correlation of $f_{\{a,a,J\}}[n]$ and $w[n]$,

$$\hat{\epsilon}(a, J) \;=\; \sum_n f_{\{a,a,J\}}[n]w[n]. \tag{6.56}$$

In an implementation, this would not be an on-line operation but rather a precomputation indicating values of $a$ and $J$ to be used in the parameter set of the composite dictionary. Interestingly, this precomputation itself resembles a matching pursuit. Note that the values of $a$ and $J$ for the $f_{\{a,a,J\}}[n]$ in the composite dictionary are based on the scales of symmetric behavior to be included in the dictionary. Presumably, closed form solutions for $a$ and $J$ can be found for some particular windows; such solutions are of course limited by the requirement that $J$ be an integer. The intent of this treatment, however, is not to investigate the computational issue of window matching *per se*, but instead to provide an existence proof that symmetric atoms constructed from one-sided exponentials by simple operations can reasonably mimic Gabor atoms based on standard symmetric windows. Figure 6.7 shows an example of a composite atom that roughly matches a Hanning window and a Gaussian window.

The upshot of the preceding discussion is that a composite dictionary containing a wide range of symmetric and asymmetric atoms can be constructed from uniform dictionaries of causal and anticausal damped sinusoids. Atoms resembling common symmetric Gabor atoms can readily be generated, meaning that this approach can be tailored to include standard symmetric atoms as a dictionary subset; there is no generality lost by

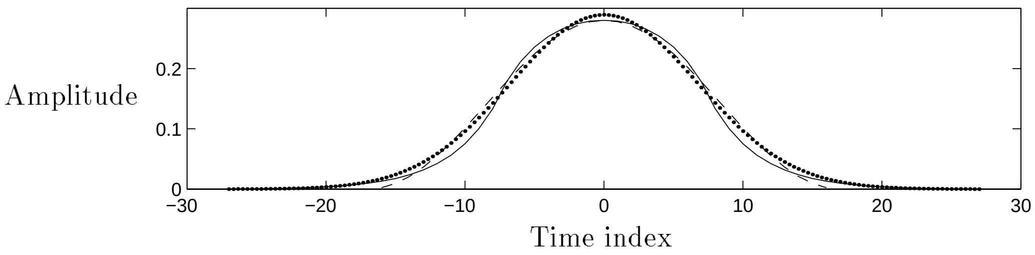

FIGURE 6.7: Symmetric composite atoms: An example of a smoothed composite atom (solid) that roughly matches a Hanning window (dashed) and a Gaussian window (dotted).

constructing atoms in this fashion. As will be shown in Section 6.4.1, the pursuit computation for dictionaries of damped sinusoids are of low cost; this leads to the methods of Section 6.4.2, namely low-cost algorithms for matching pursuit based on a composite dictionary. Such dictionaries will be discussed throughout the remainder of this chapter.

### 6.3.4 Signal Modeling

In atomic modeling by matching pursuit, the characteristics of the signal estimate fundamentally depend on the structure of the time-frequency dictionary used in the pursuit. Consider the model given in Figure 6.8, which is derived by matching pursuit with a dictionary of symmetric Gabor atoms. In the early stages of the pursuit, the algorithm arrives at smooth estimates of the global signal behavior because the large-scale dictionary elements to choose from are themselves smooth functions. At later stages, the algorithm chooses atoms of smaller scale to refine the estimate; for instance, small-scale atoms are incorporated to remove pre-echo artifacts.

In the example of Figure 6.9, the model is derived by matching pursuit with a dictionary of damped sinusoids. Here, the early estimates have sharp edges since the dictionary elements are one-sided functions. In later stages, edges that require smoothing are refined by inclusion of overlapping atoms in the model; also, as in the symmetric atom case, atoms of small scale are chosen in late stages to counteract any inaccuracies brought about by the early atoms.

In the examples of Figures 6.8 and 6.9, the dictionaries are designed for a fair comparison. Specifically, the dictionary atoms have comparable scales, and the dictionaries are structured such that the mean-squared errors of the respective atomic models have similar convergence properties. A comparison of the convergence behaviors is given in Figure 6.10(a); the plot in Figure 6.10(b) shows the energy of the pre-echo in the symmetric Gabor model and indicates that the pursuit devotes atoms at later stages to remove the pre-echo artifact. The model based on damped sinusoids does not introduce a pre-echo.

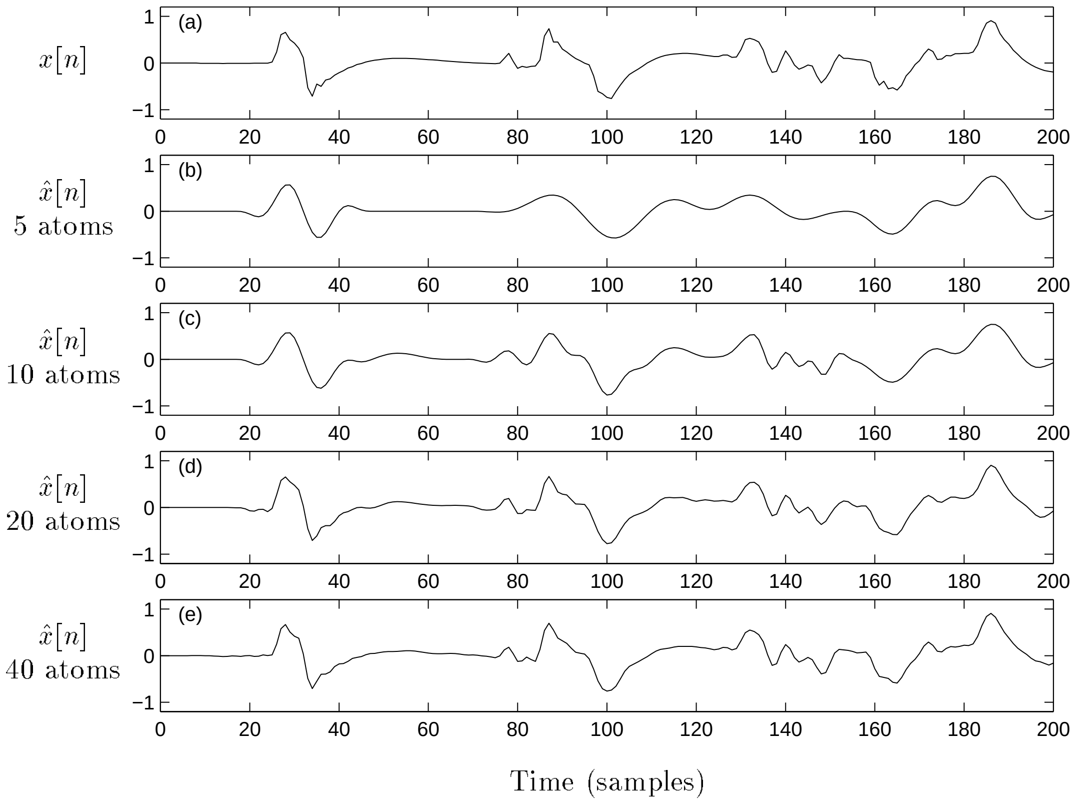

FIGURE 6.8: Signal modeling with symmetric Gabor atoms. The original signal in (a), which is the onset of a gong strike, is modeled by matching pursuit with a dictionary of symmetric Gabor atoms derived from a Hanning prototype. Approximate reconstructions at various pursuit stages are given: (b) 5 atoms, (c) 10 atoms, (d) 20 atoms, and (e) 40 atoms.

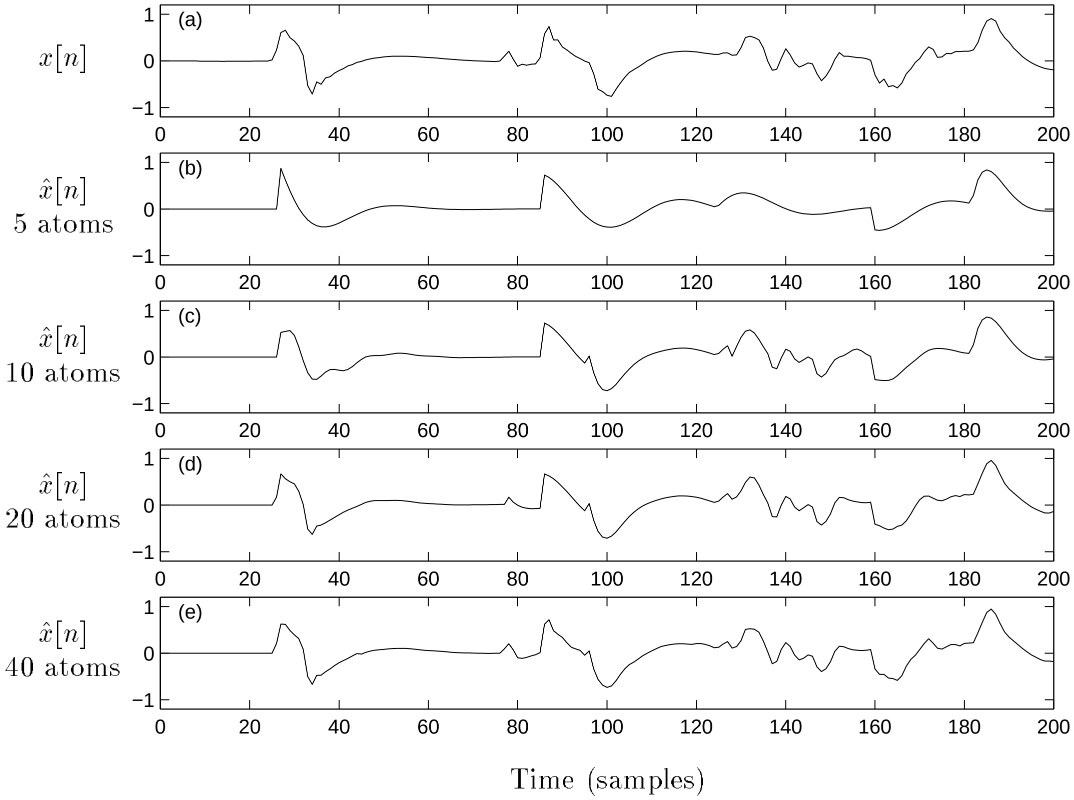

FIGURE 6.9: Signal modeling with damped sinusoidal atoms. The signal in (a), which is the onset of a gong strike, is modeled by matching pursuit with a dictionary of damped sinusoids. Approximate reconstructions at various pursuit stages are given: (b) 5 atoms, (c) 10 atoms, (d) 20 atoms, and (e) 40 atoms.

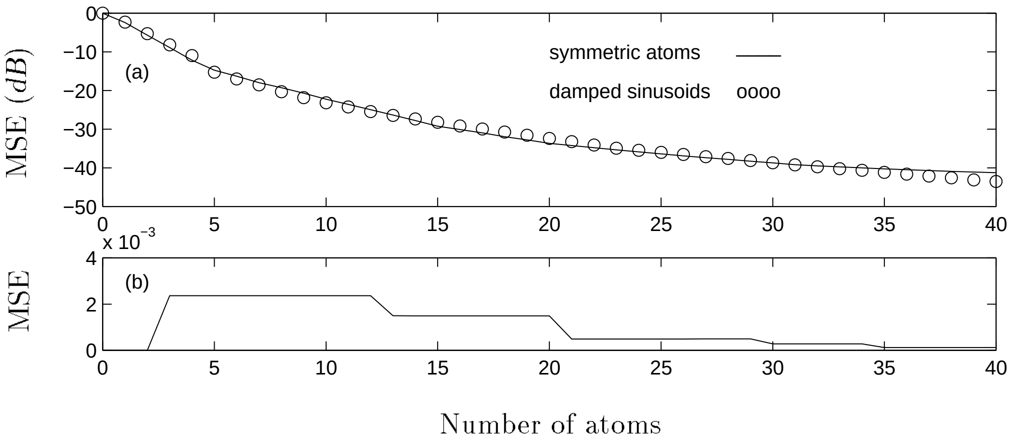

FIGURE 6.10: Mean-squared convergence of atomic models. Plot (a) shows the mean-squared error of the atomic models depicted in Figures 6.8 and 6.9. The dictionaries of symmetric Gabor atoms (solid) and damped sinusoids (circles) are designed to have similar mean-squared convergence for the signal in question. Plot (b) shows the mean-squared energy in the pre-echo of the symmetric Gabor model; the pursuit devotes atoms at later stages to reduce the pre-echo energy. The damped sinusoidal decomposition does not introduce pre-echo.

Modeling with a composite dictionary is depicted in Figure 6.11. The dictionary used here contains the same causal damped sinusoids as in the example of Figure 6.9, plus an equal number of anticausal damped sinusoids and a few smoothing orders. As will be seen in Sections 6.4 and 6.4.3, deriving the correlations with the underlying damped sinusoids is the major factor in the computational cost of pursuit with composite atoms. Compared to the pursuit based on damped sinusoids discussed earlier, then, the composite atom model shown here requires roughly twice the computation; as shown in Figure 6.12, however, this additional computation leads to a lower mean-squared error for the model. Noting further that the parameter set for composite atoms is larger than that for simple damped sinusoids or Gabor atoms, it is clear that fully comparing this composite model to the earlier models requires computation–rate–distortion considerations such as those described briefly in Section 6.2.4.

## 6.4 Computation Using Recursive Filter Banks

For arbitrary dictionaries, the cost of the matching pursuit iteration can be reduced using the correlation update relationship in Equation (6.15). For dictionaries consisting of damped sinusoids or composite atoms constructed as described in Section 6.3.3, the correlation computation for the pursuit can be carried out with simple recursive filter banks. This framework is developed in the following two sections; in Section 6.4.3, the

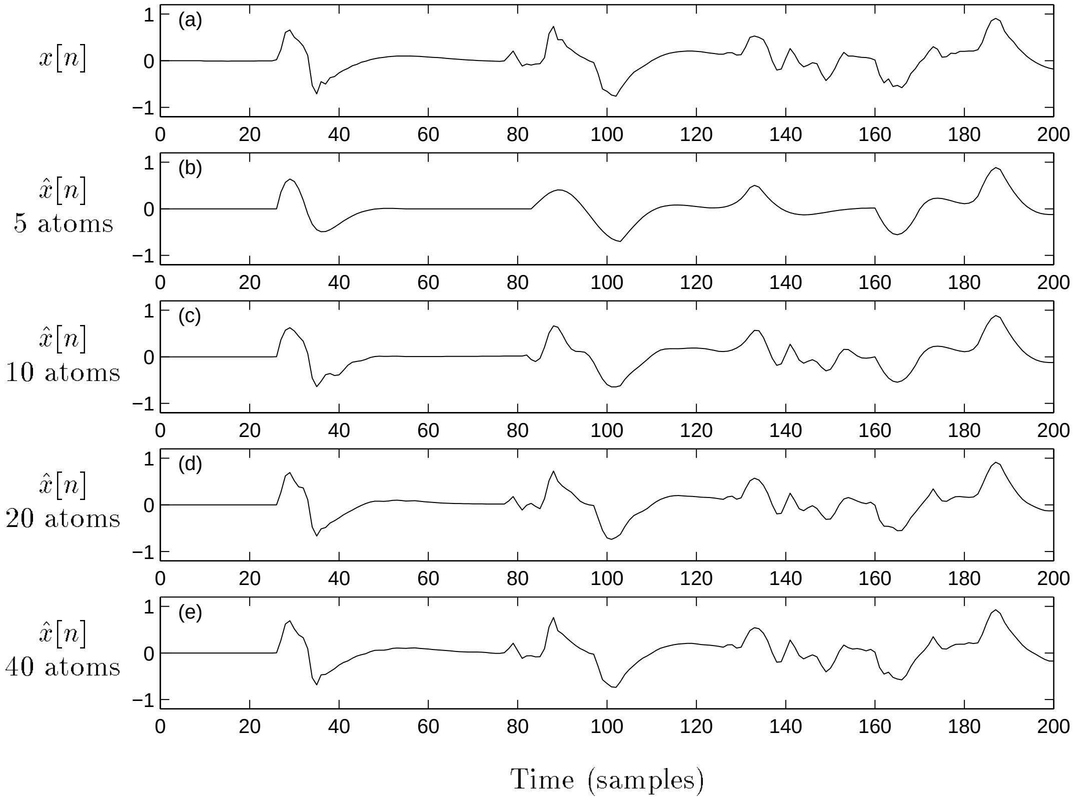

FIGURE 6.11: Signal modeling with composite atoms. The signal in (a), which is the onset of a gong strike, is modeled by matching pursuit with a dictionary of composite atoms. Approximate reconstructions at various pursuit stages are given: (b) 5 atoms, (c) 10 atoms, (d) 20 atoms, and (e) 40 atoms. The composite dictionary contains the same causal damped sinusoids as those used in the example of Figure 6.9, plus an equal number of anticausal damped sinusoids and a small number of smoothing orders.

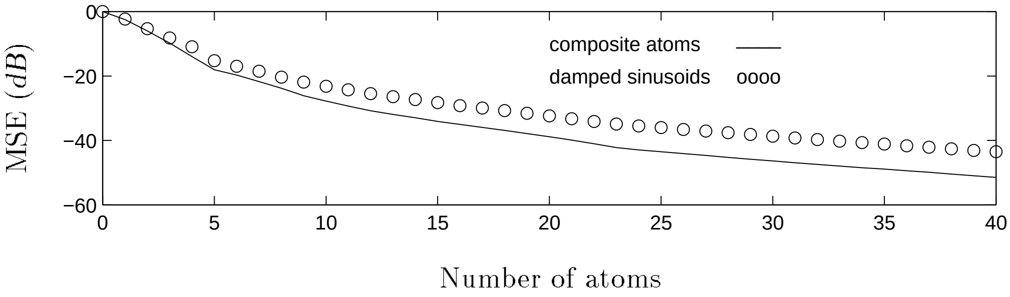

FIGURE 6.12: The mean-squared error of an atomic model using composite atoms (solid) and the mean-squared error of a model based on only the underlying causal damped sinusoids (circles). This plot corresponds to the composite atomic models given in Figure 6.11 and the damped sinusoidal decompositions of Figure 6.9.

computation requirements of the filter bank approach and the correlation update method are compared.

### 6.4.1 Pursuit of Damped Sinusoidal Atoms

For dictionaries of complex damped sinusoids, the atomic structure can be exploited to simplify the correlation computation irrespective of the update formula in Equation (6.15). It is shown here that the computation over the time and frequency parameters can be carried out with simple recursive filter banks and FFTs.

#### Correlation with complex damped sinusoids

In matching pursuit using a dictionary of complex damped sinusoids, correlations must be computed for every combination of damping factor, modulation frequency, and time shift. The correlation of a signal $x[n]$ with a causal atom $g^{+}_{\{a,\omega,\tau\}}[n]$ is given by

$$\eta_{+}(a,\omega,\tau) \;=\; S_a \sum_{n=\tau}^{\tau+L-1} x[n]\; a^{(n-\tau)} e^{-j\omega(n-\tau)}, \tag{6.57}$$

where the atoms are truncated to a length $L$ that is a function of the damping factor $a$ as described in Section 6.3.2. In the following, correlations with unnormalized atoms will be used to simplify the notation:

$$\rho_{+}(a,\omega,\tau) \;=\; \sum_{n=\tau}^{\tau+L-1} x[n]\; a^{(n-\tau)} e^{-j\omega(n-\tau)} \tag{6.58}$$

$$\;=\; \frac{\eta_{+}(a,\omega,\tau)}{S_a}. \tag{6.59}$$

Furthermore, formulating the algorithm in terms of unnormalized atoms will serve to reduce the cost of the algorithm developed in Section 6.4.2 for pursuing composite atoms.

The structure of the correlations in Equations (6.57) and (6.58) allows for a substantial reduction of the computation requirements with respect to the time shift and modulation parameters. These are discussed in turn below. Note that the correlation uses the atoms defined in Equation (6.42), in which the modulation is phase-referenced to $\tau$; alternate results related to the atoms with dereferenced modulation given in Equation (6.43) will be reviewed later.

### Time-domain simplification

The exponential structure of the atoms can be used to reduce the cost of the correlation computation over the time index; correlations at neighboring times are related by a simple recursion:

$$\rho_+(a,\omega,\tau-1) \;=\; ae^{-j\omega}\rho_+(a,\omega,\tau) \;+x[\tau-1] \;-\; a^L e^{-j\omega L}x[\tau+L-1]. \tag{6.60}$$

This is just a one-pole filter with a correction to account for truncation. If truncation effects are ignored, which is reasonable for small truncation thresholds, the formula becomes

$$\rho_+(a,\omega,\tau-1) \;=\; ae^{-j\omega}\rho_+(a,\omega,\tau) \;\;+\;\; x[\tau-1]. \tag{6.61}$$

Note that this equation is operated in reversed time to make the recursion stable for causal damped sinusoids; the similar forward recursion is unstable for $a<1$. For anticausal atoms, the correlations are given by the recursion

$$\rho_-(b,\omega,\tau+1) \;=\; be^{j\omega}\rho_-(b,\omega,\tau) \;+x[\tau+1] \;-\; b^L e^{j\omega L}x[\tau-L+1], \tag{6.62}$$

or, if truncation is neglected,

$$\rho_-(b,\omega,\tau+1) \;=\; be^{j\omega}\rho_-(b,\omega,\tau) \;+\; x[\tau+1]. \tag{6.63}$$

These recursions are operated in forward time for the sake of stability.

The equivalence of Equations (6.61) and (6.63) to filtering operations suggests interpreting the correlation computation over all possible parameters $\{a_i,\omega_i,\tau_i\}$ as an application of the signal to a dense grid of one-pole filters in the $z$-plane, which are the *matched filters* for the dictionary atoms. The filter outputs are the correlations needed for the matching pursuit; the maximally correlated atom is directly indicated by the maximum magnitude output of the filter bank. Of course, pursuit based on arbitrary atoms can be interpreted in terms of matched filters. In the general case, however, this interpretation is not particularly useful; here, it provides a framework for reducing the required computation. It should be noted that the dictionary atoms themselves correspond to the impulse responses of a grid of one-pole filters; as in the wavelet filter bank case, then, the atomic synthesis can be interpreted as an application of the expansion coefficients to a

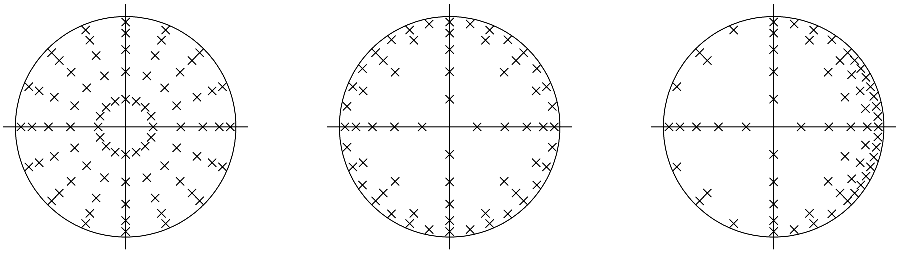

FIGURE 6.13: Filter bank interpretation and dictionary structures. The atoms in a dictionary of damped sinusoids correspond to the impulse responses of a bank of one-pole filters; for decaying causal atoms, the poles are inside the unit circle. These dictionaries can be structured in various ways as depicted above. The correlations in the pursuit are computed by the corresponding matched filters, which are time-reversed and thus have poles outside the unit circle.

synthesis filter bank. A depiction of the $z$-plane interpretation of several damped sinusoidal dictionaries is given in Figure 6.13; the dictionaries are structured for various tradeoffs in time-frequency resolution.

A recursion similar to Equation (6.60) can be written for the general case of correlations separated by an arbitrary lag $\Delta$:

$$\begin{aligned} \rho_+(a,\omega,\tau-\Delta) &= a^{\Delta}e^{-j\omega\Delta}\rho_+(a,\omega,\tau) \qquad (6.64) \\ &+\sum_{n=0}^{\Delta-1} x[n+\tau-\Delta]\; a^n\; e^{-j\omega n} \;-\; a^L e^{-j\omega L}\;\sum_{n=0}^{\Delta-1} x[n+\tau-\Delta+L]\; a^n\; e^{-j\omega n}. \end{aligned}$$

For $w = 2\pi k/K$, the last two terms can be computed using the DFT:

$$\begin{aligned} \rho_+(a,2\pi k/K,\tau-\Delta) &= a^{\Delta}e^{-j\omega\Delta}\rho_+(a,\omega,\tau) \qquad (6.65) \\ &+ \mathrm{DFT}_K\left\{x[n+\tau-\Delta]a^n\right\}|_k \;-\; a^L e^{-j\omega L}\;\mathrm{DFT}_K\left\{x[n+\tau-\Delta+L]a^n\right\}|_k, \end{aligned}$$

where $n \in [0, \Delta-1]$ in the latter terms, which could be combined into a single DFT. If truncation effects are ignored, the second DFT term is neglected and the relationship is again more straightforward. Similar simplifications have been reported in the literature for short-time Fourier transforms using one-sided exponential windows [228] as well as more general cases [229].

### Frequency-domain simplification

A simplification of the correlation computations across the frequency parameter can be achieved if the $z$-plane filter bank, or equivalently the matching pursuit dictionary, is structured such that the modulation frequencies are equi-spaced for each damping factor.

If the filters (atoms) are equi-spaced angularly on circles in the $z$-plane, the discrete Fourier transform can be used for the computation over $\omega$. For $\omega = 2\pi k/K$, the correlation is given by

$$\begin{aligned} \rho_+(a, 2\pi k/K, \tau) &= \sum_{n=0}^{L-1} x[n+\tau] a^n e^{-j2\pi kn/K} & (6.66) \\ &= \mathrm{DFT}_K \left\{ x[n+\tau] \; a^n \right\} \big|_k , & (6.67) \end{aligned}$$

where $n \in [0, L-1]$ and $K \geq L$. Thus, FFT algorithms can be used to compute correlations over the frequency index. Note that such an FFT-based simplification can be applied to any dictionary of harmonically modulated atoms.

At a fixed scale, correlations must be computed at every time-frequency pair in the index set. There are two ways to cover this time-frequency index plane; these correspond to the dual interpretations of the STFT depicted in Figure 2.1. The first approach is to carry out a running DFT with an exponential window; windowing and the DFT require $L$ and $K \log K$ multiplies per time point, respectively, so this method requires roughly $N(L + K \log K)$ real multiplies for a signal of length $N$. The second approach is to use a DFT to initialize the $K$ matched filters across frequency and then compute the outputs of the filters to evaluate the correlations across time; indeed, the signal can be zero-padded such that the filters are initialized with zero values and no DFT is required. Recalling the recursion of Equation (6.60), this latter method requires one complex multiply and one real-complex multiply per filter for each time point, so it requires $6KN$ real multiplies, $2KN$ of which account for truncation effects and are not imperative. For large values of $K$, this is significantly less than the multiply count for the running DFT approach, so the matched filter approach is the method of choice.

### Results for dereferenced modulation

The results given in the previous sections hold for an atom whose modulation is referenced to the time origin of the atom as in Equations (6.39), (6.42), and (6.44). This local time reference has been adhered to since it allows for an immediate filter bank interpretation of the matching pursuit analysis; also, synthesis based on such atoms can be directly carried out using recursive filters. For the construction and pursuit of composite atoms, however, the dereferenced atoms defined in Equations (6.41), (6.43), and (6.45) are of importance. The correlation formulae for dereferenced damped sinusoids can be derived by combining the relation in Equation (6.41) with the expression in Equation (6.58) to arrive at:

$$\tilde{\rho}_+(a, \omega, \tau) = e^{-j\omega\tau} \rho_+(a, \omega, \tau), \qquad (6.68)$$

so Equations (6.61) and (6.63) can be reformulated as

$$\tilde{\rho}_+(a, \omega, \tau - 1) = a\tilde{\rho}_+(a, \omega, \tau) + e^{-j\omega(\tau-1)} x[\tau - 1] \qquad (6.69)$$

$$\tilde{\rho}_-(b,\omega,\tau+1) \;=\; b\tilde{\rho}_+(b,\omega,\tau) + e^{-j\omega(\tau+1)}x[\tau+1]. \tag{6.70}$$

When the modulation depends on the atomic time origin, the pursuit can be interpreted in terms of a modulated filter bank; for dereferenced modulation, however, the equivalent filter bank has a heterodyne structure. This distinction was discussed at length with respect to the STFT in Section 2.2.1. As will be seen in Section 6.4.2, dereferencing the modulation simplifies the relationship between the signal correlations with composite atoms and the correlations with underlying damped sinusoids; for this reason, future considerations will be focus primarily on the case of dereferenced modulation.

### Real decompositions of real signals

If dictionaries of complex atoms are used in matching pursuit, the correlations and hence the expansion coefficients for signal decompositions will generally be complex; a given coefficient thus provides both a magnitude and a phase for the atom in the expansion. For real signals, decomposition in terms of complex atoms can be misleading. For instance, for a signal that consists of one real damped sinusoid, the pursuit does not simply find the constituent conjugate pair of atoms as might be expected; this occurs because an atom and its conjugate are not orthogonal. For real signals, then, it is preferable to consider expansions in terms of real atoms:

$$\bar{g}^+_{\{a,\omega,\tau,\phi\}} \;=\; \bar{S}_{\{a,\omega,\phi\}}a^{(n-\tau)}\cos\left[\omega(n-\tau)+\phi\right]u[n-\tau], \tag{6.71}$$

or, in the case of dereferenced modulation,

$$\tilde{\bar{g}}^+_{\{a,\omega,\tau,\phi\}} \;=\; \tilde{\bar{S}}_{\{a,\omega,\tau,\phi\}}a^{(n-\tau)}\cos\left[\omega n+\phi\right]u[n-\tau]. \tag{6.72}$$

The two cases differ by a phase offset which affects the unit-norm scaling as well as the modulation.

In the case of a complex dictionary, the atoms are indexed by the three parameters $\{a,\omega,\tau\}$ and the phase of an atom in the expansion is given by its correlation. In contrast, a real dictionary requires the phase parameter as an additional index because of the explicit presence of the phase in the argument of the cosine in the atom definition. The phase is not supplied by the correlation computation as in the complex case; like the other parameters, it must be discretized and incorporated as a dictionary parameter in the pursuit, which results in a larger dictionary and thus a more complicated search. Furthermore, the correlation computations are more difficult than in the complex case because the recursion formulae derived earlier do not apply for these real atoms. These problems can be circumvented by using a complex dictionary and considering conjugate subspaces according to the formulation of Section 6.2.3.

Conjugate subspace pursuit can be used to search for conjugate pairs of complex damped sinusoids; the derivation leading to Equation (6.21) verifies that this approach will arrive at a decomposition in terms of real damped sinusoids if the original signal is real. The advantage of this method is indicated by Equations (6.19) and (6.20), which show that the expansion coefficients and the maximization metric in the conjugate pursuit are both functions of the correlation of the residual with the underlying complex atoms; this means that the computational simplifications for a dictionary of complex damped sinusoids can be readily applied to calculation of a real expansion. The real decomposition found by this approach, expressed in the general case in Equation (6.23), can be written explicitly as

$$x[n] \approx 2 \sum_{i=1}^{I} S_{a_i} A_i a_i^{(n-\tau_i)} \cos\left[\omega_i n + \phi_i\right], \tag{6.73}$$

where $A_i e^{j\phi_i} = \alpha_i(1)$ and the modulation is dereferenced. As in the complex case, the phases of the atoms in this real decomposition are provided directly by the computation of the expansion coefficients; the phase is not required as a dictionary index, *i.e.* an explicit search over a phase index is not required in the pursuit. By considering signal expansions in terms of conjugate pairs, the advantages of the complex dictionary are fully maintained; furthermore, note that the dictionary for the conjugate search is effectively half the size of the full complex dictionary since atoms are considered in conjugate pairs.

It is important to note that Equation (6.73) neglects the inclusion of unmodulated exponentials in the signal expansion. Such atoms are indeed present in the complex dictionary, and all of the recursion speedups apply trivially; furthermore, the correlation of an unmodulated atom with a real signal is always real, so there are no phase issues to be concerned with. An important caveat, however, is that the conjugate pursuit algorithm breaks down if the atom is purely real; the pursuit requires that the atom and its conjugate be linearly independent, meaning that the atom must have nonzero real and imaginary parts. Thus, a fix is required if real unmodulated exponentials are to be admitted in the signal model. The $i$-th stage of the fixed algorithm is as follows: first, the correlations $\beta = \langle g, r_i \rangle$ for the entire dictionary of complex atoms are computed using the simplifications described. Then, energy minimization metrics for both types of atoms are computed and stored: for real atoms, the metric is $|\beta|^2$ as indicated in Equation (6.12); for conjugate subspaces, the metric is $\beta^*\alpha(1) + \beta\alpha(1)^*$ as given in Equation (6.20), where $\alpha(1)$ is as defined in Equation (6.19) and $, = \langle g, g^* \rangle$ can be expressed as

$$, (a, \omega) = S_a^2 \left( \frac{1 - a^{2L} e^{-j2\omega L}}{1 - a^2 e^{-j2\omega}} \right). \tag{6.74}$$

These metrics quantify the amount of energy removed from the residual in the two cases; maximization over these metrics indicates which real component should be added to the signal expansion at the $i$-th stage to minimize the energy of the new residual $r_{i+1}[n]$.

As a final comment on real decompositions, it is interesting to note that the description of a signal in terms of conjugate pairs does not require more data than a model using complex atoms. Either case simply requires the indices $\{a, \omega, \tau\}$ and the complex number $\alpha(1)$ for each atom in the decomposition. There is of course additional computation in both the analysis and the synthesis in the case of conjugate pairs. As discussed above, this improves the ability to model real signals; in a sense, this improvement arises because the added computation enables the model data to encompass twice as many atoms in the conjugate pair case as in the complex case.

### 6.4.2 Pursuit of Composite Atoms

Using matching pursuit to derive a signal model based on composite atoms requires computation of the correlations of the signal with these atoms. Recalling the form of the composite atoms given in Equations (6.51) and (6.52), these correlations have, by construction, a simple relationship to the correlations with the underlying one-sided atoms:

$$\begin{aligned}
\tilde{\rho}(a, b, J, \omega, \tau) &= S_{\{a,b,J\}} \sum_{\Delta=0}^{J-1} \left[ \frac{\tilde{\eta}_+(a, \omega, \tau+\Delta)}{S_a} + \frac{\tilde{\eta}_-(b, \omega, \tau+\Delta)}{S_b} - x[\tau+\Delta] \right] && (6.75)\\
&= S_{\{a,b,J\}} \sum_{\Delta=0}^{J-1} \left[ \tilde{\rho}_+(a, \omega, \tau+\Delta) + \tilde{\rho}_-(b, \omega, \tau+\Delta) - x[\tau+\Delta] \right]. && (6.76)
\end{aligned}$$

The correlation with any hybrid atom can thus be computed based on the correlations derived by the recursive filter banks discussed earlier; this computation is most straightforward if dereferenced modulation is used in the constituent atoms and if these underlying atoms are unnormalized. Essentially, any atom constructed according to Equation (6.52), which includes simple damped sinusoids, can be added to the modeling dictionary at the cost of one multiply per atom to account for scaling. Computation is discussed further in Section 6.4.3.

For composite atoms, real decompositions of real signals take the form

$$x[n] \approx 2 \sum_{i=1}^{I} A_i f_{\{a_i, b_i, J_i\}}[n - \tau_i] \cos\left(\omega_i n + \phi_i\right), \qquad (6.77)$$

where $f_{\{a_i, b_i, J_i\}}$ is as defined in Equation (6.46) and $A_i e^{j\phi_i} = \alpha_i(1)$ from Equation (6.19).

### 6.4.3 Computation Considerations

This section compares the computational cost of two matching pursuit implementations: pursuit based on correlation updates [38] and pursuit based on recursive filter banks. In this comparison, the cost is measured in terms of memory requirements and multiplicative operations. Simple search operations, table lookups, and conditionals are neglected in the cost measure. Furthermore, computation before the first iteration of either algorithm is allowed without a direct penalty; precomputation is considered only with respect to the amount of memory required to store precomputed data. Startup cost for the first iteration is considered separately; in cases where only a few atoms are to be derived, the startup arithmetic in the update algorithm may constitute an appreciable percentage of the overall computation. The results of these considerations are summarized in Table 6.1.

### Notation

In the following comparisons, the signal is assumed to be real and of length $N$. A composite dictionary based on damped sinusoids will be considered. The index set for this dictionary will consist of $A$ different causal damping factors, $B$ anticausal damping factors, $H$ smoothing orders, $K$ modulations, and $N$ time shifts, meaning that the dictionary has $M = ABHKN$ atoms; using $S$ to denote the number of scales present, namely $S = ABH$, the size of the dictionary can be expressed as $M = SKN$. The average scale or atom length will be denoted by $L$; note that for atoms having average time support $L$, the correlation $\langle g, x\rangle$ requires $L$ real-complex multiplies on average. The following considerations of the two matching pursuit algorithms focus on pursuit of complex atoms since the evaluation of a real model based on a complex pursuit has equal cost in both implementations; note also that deriving the correlation magnitudes requires the same amount of computation in both approaches. The relevant point of comparison is the computation required to calculate $\langle g, r_i\rangle$ for all of the complex atoms $g \in D$ at some stage $i$ of the algorithm. The treatment in this section depends on the damped sinusoidal structure, but this is not a limiting restriction since composite atoms with a wide range of time-frequency behaviors can be constructed based on damped sinusoids.

### Precomputation in the update algorithm

To derive the correlations $\langle g, r_{i+1}\rangle$ at stage $i+1$ of the pursuit, the update approach uses the equation

$$\langle g, r_{i+1}\rangle \;=\; \langle g, r_i\rangle \;-\; \alpha_i\langle g, g_i\rangle, \tag{6.78}$$

which relates the correlations at stage $i+1$ to those computed at stage $i$. The update method thus relies on precomputation and storage of the dictionary cross-correlations $\langle g, g_i\rangle$ to reduce the computational cost of the pursuit. If this storage is done without taking the sparsity or redundancy of the data into account, $M^2$ cross-correlations must be stored.

A simple example shows that the brute force approach to cross-correlation storage is prohibitive. Consider analysis of a $10ms$ frame of high-quality audio consisting of roughly $N = 400$ samples. In a rather small dictionary with $K = 32$, $A = 10$, $B = 1$, and $H = 1$, there are roughly $M = 10^5$ atoms. Storage of the complex cross-correlations then requires $2M^2 = 2 \times 10^{10}$ memory locations. This is altogether unreasonable, so it is necessary to investigate the possibility of memory-computation tradeoffs; such tradeoffs occur commonly in algorithm design.

The memory requirement can be relaxed by considering the sparsity and redundancy of the dictionary cross-correlation data. First, many of the atom pairs have no time overlap and thus zero correlation; these cases can be handled with conditionals. For atoms that do overlap, the correlation storage can be reduced using the following formulation. Introducing the simplifying notation

$$\begin{aligned} g(s_0, \omega_0, \tau_0) &= g_{\{a_0,b_0,J_0,\omega_0,\tau_0\}}[n] = f_{\{s_0\}}[n-\tau_0]e^{j\omega_0 n} && (6.79)\\ g(s_1, \omega_1, \tau_1) &= g_{\{a_1,b_1,J_1,\omega_1,\tau_1\}}[n] = f_{\{s_1\}}[n-\tau_1]e^{j\omega_1 n}, && (6.80)\end{aligned}$$

where $s_0$ and $s_1$ serve as shorthand for the effective scales of the atoms and $f[n]$ is a unit-norm envelope constructed as in Equation (6.46), the cross-correlation of two composite atoms can be expressed as

$$\begin{aligned} \langle g(s_0,\omega_0,\tau_0), g(s_1,\omega_1,\tau_1)\rangle &= \sum_n f_{\{s_0\}}[n-\tau_0] f_{\{s_1\}}[n-\tau_1] e^{j(\omega_0-\omega_1)n} && (6.81)\\ \{\text{letting } m = n-\tau_0\} \quad &= \sum_m f_{\{s_0\}}[m] f_{\{s_1\}}[m-(\tau_1-\tau_0)] e^{j(\omega_1-\omega_0)(m+\tau_0)} && (6.82)\\ &= e^{j(\omega_1-\omega_0)\tau_0} \, \langle g(s_0,0,0), g(s_1,\omega_1-\omega_0,\tau_1-\tau_0)\rangle, && (6.83)\end{aligned}$$

which shows that the cross-correlation, with the exception of a phase shift, does not depend on the absolute time locations of the atoms but rather on their relative locations. Also, the correlation is only a function of the frequency difference; moreover, it only depends on the absolute difference since negative values of $\omega_1 - \omega_0$ can be accounted for by conjugation:

$$\langle g(s_0,0,0), g(s_1,\omega_1-\omega_0,\tau_1-\tau_0)\rangle = \langle g(s_0,0,0), g(s_1,\omega_0-\omega_1,\tau_1-\tau_0)\rangle^* . \qquad (6.84)$$

Beyond these simplifications, there is also redundancy in the cross-correlations for scale pairs:

$$\begin{aligned} \langle g(s_1,\omega_1,\tau_1), g(s_0,\omega_0,\tau_0)\rangle &= \langle g(s_0,\omega_0,\tau_0), g(s_1,\omega_1,\tau_1)\rangle^* && (6.85)\\ &= e^{j(\omega_0-\omega_1)\tau_0} \, \langle g(s_0,0,0), g(s_1,\omega_1-\omega_0,\tau_1-\tau_0)\rangle^*. && (6.86)\end{aligned}$$

This relationship can be exploited to reduce the memory requirements by roughly a factor of two.

The formulations given above drastically reduce the amount of memory required to store the dictionary cross-correlations. With regards to the modulation frequencies, there are $K$ distinct possibilities for $|\omega_1 - \omega_0|$. With regards to the time shifts, the $S$ different scales in the dictionary can be considered in pairs using $L$ to approximate the number of lags that lead to overlap and nonzero correlation; there are roughly $S^2L$ different configurations. In total, then, $2S^2KL$ memory locations are required to store the distinct cross-correlation values; taking the scale-pair redundancy into account, this count is reduced to $S^2KL$. For the simple audio example discussed above, this amounts to roughly $6 \times 10^4$ locations for $L = 20$. Noting the phase shift in Equation (6.83), this reduction in the memory requirements introduces a complex multiply, or three real multiplies, for each correlation update.[2]

### Precomputation in the filter bank algorithm

In the filter bank approach, the pursuit computation is based on correlations with unnormalized atoms as formalized in Equation (6.76), which holds for the general case of a composite dictionary as well as for the limiting case of a dictionary of damped sinusoids, where $J = 1$ and $b = 0$. This correlation computation requires scaling by $S_{\{a,b,J\}}$. To reduce the amount of computation per iteration, these scaling factors can be precomputed and stored. The cost of this precomputation is not of particular interest here; the important issue is the amount of memory required to store the precomputed values. In the general case where the values of the damping factors $a$ and $b$ do not exhibit any particular symmetry, storing the scaling factors requires $S = ABH$ memory locations.

### The first iteration of the update algorithm

In the first stage of the update algorithm, all of the correlations with the dictionary atoms must be computed. This exhaustive computation requires $ML = SKL$ real-complex multiplies, or $2ML$ real multiplies. Of course, this computation can be carried out with recursive filter banks at a lower cost, but such a merged approach will not be treated here. In any event, these complex correlations must be stored, which requires $2M$ memory locations. The total memory needed in the update algorithm is then $S^2KL + 2M$. Note that the signal is needed in the first stage of the algorithm but is not required thereafter.

### Later iterations of the update algorithm

Once the dictionary cross-correlations have been precomputed and the correlations for the first stage of the pursuit have been calculated and stored, the cost of the update algorithm depends only on the update formula. Each stage of the algorithm involves $M$ complex-complex multiplies ($3M$ real) to multiply the $M$ cross-correlations by $\alpha_i$, plus another $M$ complex-complex multiplies to carry out the phase shift given in

[2] The complex multiply $(a+bj)(c+dj) = ac - bd + j(ad+bc)$ can be carried out using three multiplies by computing $c(a+b), b(c+d)$, and $d(a-b)$. Then, $ac-bd$ is given by the difference of the first two terms; $ad+bc$ is the sum of the second and the third terms.

Equation (6.83), for a total of $6M$ real multiplies per iteration. Note that in the update algorithm it is not necessary to keep the signal in memory after the first iteration or to ever actually compute the residual signal.

### Iterations in the filter bank approach

In matching pursuit based on recursive filter banks, the scaling factors $S_{\{a,b,J\}}$ are precomputed and available via lookup. In addition to the scaling factors, the residual signal must be stored in this implementation; this requires $N$ memory locations. The final memory requirement is that in order to evaluate correlations with composite atoms, correlations with the constituent unnormalized damped sinusoids must be stored. For a smoothing order of $J$, this requires correlations with $J$ causal and $J$ anticausal damped sinusoids. Storing these underlying correlations in a local manner thus requires $2(A+B)KJ$ locations; global storage requires $2(A+B)KN$ locations. Note that the memory cost is scaled by a factor of two since the correlations are complex numbers. The worst case memory requirement in the filter bank case is then $S+N+2(A+B)KN$.

With regards to computation, the algorithm uses $(A+B)K$ recursive filters to derive the correlations. In the dereferenced case given in Equations (6.69) and (6.69), each recursion requires four real-real multiplies for each of the $N$ time points if atom truncation is neglected; the count increases to six if truncation is included. As indicated in Equation (6.76), correlations with composite atoms are computed by adding the correlations with constituent unnormalized damped sinusoids and then scaling with the appropriate factor; this construction process introduces $S = ABH$ real-complex multiplies, or $2S$ real multiplies. Thus, $6(A+B)KN+ABH$ real multiplies are needed to compute the pursuit correlations. Once an atom is chosen based on these correlations, the residual must be updated; this requires roughly $5L$ multiplies to generate the unit-norm atomic envelope, modulate it to the proper frequency, and weight it with its expansion coefficient prior to subtraction from the signal. The total computational cost per iteration for the filter bank algorithm is thus $5L+6(A+B)KN+2S$.

## 6.5 Conclusions

Atomic models provide descriptions of signals in terms of localized time-frequency events. Derivation of optimal models based on overcomplete sets of atoms is computationally prohibitive, but effective models can be arrived at by greedy algorithms such as matching pursuit and its variations. In this chapter, matching pursuit was developed as an approach for deriving compact signal-adaptive parametric models based on dictionaries of time-frequency atoms. Time-frequency dictionaries consisting of symmetric Gabor atoms, damped sinusoids, and composite atoms constructed from underlying damped sinusoids

| | MEMORY<br>(real numbers) | | COMPUTATION<br>(real multiplies) | |
|---|---|---|---|---|
| Method | Precomp. | Algorithm | First iteration | Later iterations |
| Update | $S^2KL$ | $2M$<br>$= 2ABHKN$ | $2ML$<br>$= 2ABHKNL$ | $6M$<br>$= 6ABHKN$ |
| Filter bank | $S$ | $N + 2(A + B)KN$ | $5L + 2ABH + 6(A + B)KN$ | |

TABLE 6.1: Tabulation of computation considerations: memory and computation requirements for matching pursuit using the update algorithm and the recursive filter bank method. $N$ is the length of the signal; the dictionary index set contains $A$ causal damping factors, $B$ causal damping factors, $H$ smoothing orders, $S = ABH$ scales, $K$ modulations, and $N$ time shifts, meaning that the dictionary contains $M = SKN = ABHKN$ distinct atoms. $L$ is the average time support of a dictionary atom.

were considered and compared. It was shown that the matching pursuit computation for both damped sinusoidal atoms and composite atoms can be carried out efficiently using simple recursive filter banks.

Chapter 7

# Conclusions

This thesis explores a variety of signal models, namely the sinusoidal model, multiresolution extensions of the sinusoidal model, residual models, pitch-synchronous wavelet and Fourier representations, and atomic decompositions. The key issues dealt with in this text are summarized in the following section; thereafter, directions for further research are discussed.

## 7.1 Signal-Adaptive Parametric Representations

In modeling a signal, it is of primary importance that the model be adapted to the signal in question. Otherwise, the model will not necessarily provide a meaningful or useful representation of the signal. The models considered in this thesis are examples of such signal-adaptive models. In each case, the model is constructed in a signal-adaptive fashion; this leads to compact models which are useful for analysis, compression, denoising, and modification. Some of these capabilities are enhanced by the parametric nature of the models. If a signal is represented in terms of perceptually salient parameters, meaningful modifications can be made by simple adjustment of the parameters; furthermore, perceptual principles can be readily applied to achieve data reduction. The following sections provide a review of the main issues discussed in each chapter.

### 7.1.1 The STFT and Sinusoidal Modeling

In Chapter 2, the sinusoidal model is developed as a parametric extension of the short-time Fourier transform. The filter bank interpretation of the STFT is reviewed and extended, and various perfect reconstruction criteria are developed. In Section 2.2.2, however, it is shown by a simple example that such a rigid filter bank does not provide a compact representation of an evolving signal. This motivates representing the subband signals in terms of a parametric model based on estimating and tracking evolving sinusoidal partials. Analysis methods for estimating the partial parameters are considered;

the treatment includes a linear algebraic interpretation of spectral peak picking. Also, time-domain and frequency-domain synthesis techniques are discussed.

### 7.1.2 Multiresolution Sinusoidal Modeling

If operated with a fixed frame size, the sinusoidal model has difficulties representing nonstationary signals. Accurate reconstruction of dynamic behavior can be achieved by carrying out the sinusoidal model in a multiresolution framework. In Chapter 3, two multiresolution extensions based respectively on filter banks and adaptive time segmentation are discussed; the focus is placed primarily on the latter method, which is shown to substantially mitigate pre-echo distortion. A dynamic program for deriving pseudo-optimal segmentations is developed; furthermore, globally exhaustive and simple heuristic algorithms are both considered, and the various approaches are compared with respect to computational cost.

### 7.1.3 Residual Modeling

In parametric methods such as the sinusoidal model, the analysis-synthesis process generally does not lead to a perfect reconstruction of the original signal; there is a nonzero difference between the original and the inexact reconstruction. For high-quality synthesis, it is important to model this residual and incorporate it in the signal reconstruction; this accounts for salient features such as breath noise in a flute sound. In Chapter 4, residual modeling for sinusoidal analysis-synthesis is discussed. For multiresolution sinusoidal models, the residual can be perceptually well-modeled as white noise shaped by a filter bank with time-varying channel gains whose subbands are spaced in frequency according to psychoacoustic considerations. The channel gains are determined by analyzing the residual; these gains serve as an efficient parametric representation of the residual. Strictly speaking, this residual analysis-synthesis is not signal-adaptive; however, it is necessary to consider such methods for use with near-perfect reconstruction models such as those described in this text. When used in conjunction with the sinusoidal model, this approach leads to high-fidelity reconstruction of natural sounds.

### 7.1.4 Pitch-Synchronous Representations

For pseudo-periodic signals, compaction can be achieved by incorporating the pitch in the signal model. In Chapter 5, pitch-synchronous modeling and processing is discussed. It is shown that both the sinusoidal model and the wavelet transform can be improved by pitch-synchronous operation when the original signal is pseudo-periodic. In either approach, periodic signal regions can be efficiently represented while aperiodic regions, *e.g.* note onsets, can be modeled using the perfect reconstruction capability of the underlying transform, namely the discrete wavelet transform in the pitch-synchronous wavelet case and the Fourier transform in the pitch-synchronous sinusoidal model.

### 7.1.5 Atomic Decompositions

In Chapter 3, the sinusoidal model is interpreted as a decomposition in terms of time-frequency atoms constructed according to parameters extracted from the signal; this interpretation motivates the various multiresolution extensions of the model. In Chapter 5, pitch-synchronous transforms are similarly interpreted as granulation methods; in those approaches, a pseudo-periodic signal is decomposed into pitch period grains according to estimates of the signal periodicity, and these grains are further modeled using Fourier or wavelet techniques. The atomic models discussed in Chapter 6 differ from these representations in that the atoms for the model are not derived from signal parameters; rather, parametric atoms that match the signal behavior are chosen from an overcomplete dictionary.

Atomic models based on overcomplete dictionaries of time-frequency atoms can be computed using the matching pursuit algorithm. Typically, such dictionaries consist of Gabor atoms based on a symmetric prototype window; such atoms have difficulties representing transient behavior, however. With the goal of overcoming this problem, alternative dictionaries are considered, namely dictionaries of damped sinusoids as well as dictionaries of general asymmetric atoms constructed based on underlying causal and anticausal damped sinusoids. It is shown in Section 6.4 that the matching pursuit computation for either type of atom can be carried out with low-cost recursive filter banks.

## 7.2 Research Directions

The work discussed in this thesis has a number of natural extensions. This section describes extensions in audio coding and provides suggestions for further work involving overcomplete expansions.

### 7.2.1 Audio Coding

The current standard methods in audio coding, namely MPEG and related coding schemes, use cosine-modulated filter banks; perceptual criterion are applied to the subband signals to achieve data reduction [12, 7, 9, 8]. Some signal adaptivity is achieved by adjusting the filter lengths according to the signal behavior; in terms of the prototype window for the filter bank, a short window is used in the vicinity of transients and a long window is used for stationary regions. It is an open question whether the rate-distortion performance of this industry standard can be rivaled by parametric methods such as the sinusoidal model or overcomplete atomic models.

### Sinusoidal modeling

In the sinusoidal model, which has received recent attention for the application of audio coding, quantization is a primary open issue [230, 231]. For instance, it is of interest to incorporate perceptual resolution limits in the amplitude and frequency quantization schemes. Another important psychoacoustic consideration is the formal characterization of distortion artifacts such as pre-echo; such characterizations are required if the method is to be compared to standard techniques.

For audio coding with the sinusoidal model, a number of data reduction techniques and modeling improvements are of possible interest. First, predictive models of the partial tracks in time and frequency may be useful for data reduction; linear prediction of the spectral envelope has been applied with some success to speech coding based on the sinusoidal model [232]. Such prediction may also prove useful for assisting with the estimation of sinusoidal parameters in upcoming signal frames. In this light, the sinusoidal model holds some promise for the application of audio transmission on packet-lossy networks; signal segments corresponding to lost packets can be reconstructed by using models of track evolution to interpolate the parameters from adjacent received packets.

Since compaction leads to coding gain, audio coding using the sinusoidal model would benefit from the ability to derive the most compact sinusoidal representation of a signal. Given that the expansion functions in an oversampled DFT correspond to a tight frame, methods of obtaining optimal or pseudo-optimal sparse frame expansions provide a means for obtaining such optimally compact sinusoidal models. In this sense, some improvements in sinusoidal modeling techniques may actually arise from the theory of overcomplete expansions. Note that a procedure similar to matching pursuit is used in [154] to estimate sinusoidal components; as discussed in Chapter 6, however, such pursuit does not yield an optimal compact representation, so some improvement can be achieved. In addition to improvements in compaction due to frame-theoretic approaches, further investigations along such lines may indicate successive refinement frameworks for the sinusoidal model that offer advantages over current techniques.

For audio coding based on the sinusoidal model, multiresolution methods are of significant interest for a number of reasons beyond their improved signal representation capabilities. For one, dynamic segmentation allows for optimization of the model in a rate-distortion sense, which is of course useful for coding applications; furthermore, psychoacoustic criterion such as the perceptual entropy used in MPEG can be incorporated in the dynamic program to determine the optimal segmentation. Multirate filter bank methods coupled with sinusoidal modeling are also of interest for audio coding since they allow for modeling and synthesis at subsampled rates; such efficient synthesis is of great importance given the applications of audio recording and broadcasting, both of which demand real-time signal reconstruction.

The possible advances suggested above can be viewed as steps in the development of a fully optimal sinusoidal model. In addition to an appropriate multiresolution scheme such as dynamic segmentation, achieving a fully optimal model indeed requires global consideration of the parameter estimation technique (*e.g.* spectral peak picking), the line tracking method, and the parameter interpolation functions used for reconstruction. These various components of sinusoidal analysis-synthesis are intrinsically interdependent; it is an open question as to how these dependencies can be accounted for in model optimization.

Finally, it should be noted that it is of interest in the multimedia community to carry out signal modifications in the compressed domain. Some modifications based on MPEG audio compression have been developed, but these are somewhat restricted in comparison to the rich class of modifications enabled by a sinusoidal signal model [231].

### Atomic models

Whereas there are clear indications that the sinusoidal model may be useful as an audio coding scheme, it has not yet been shown that atomic models based on overcomplete dictionaries are similarly promising. One fundamental advance required for application of atomic modeling to audio coding is the ability to carry out matching pursuit effectively in a frame-by-frame manner so that signals of arbitrary length can be processed. Matching pursuit using fixed frames has been described in the literature [220], but such an approach is unable to identify or model atomic components that overlap frame boundaries.

There are several additional noteworthy points regarding atomic models and audio coding. First, an atomic signal model would allow complex time-frequency masking principles to be incorporated in the coding scheme. Also, given an atomic model, it can be expected that some coding gain can be achieved based on the occurrence of redundant structures in the atomic index sets; entropy coding of the indices may prove useful. Finally, further capabilities for identifying basic signal behavior are of interest; for instance, pursuit of atoms with harmonic structure may prove useful for audio signal modeling.

Beyond audio coding, another conceivable application of atomic modeling is to represent the residual of some independent analysis-synthesis process such as the sinusoidal model. An analogy is the compression technique described in [181], where matching pursuit is used to derive a model of the residual in a motion-compensated video coder. In that approach, many simplifications arise due to the structure of the residual and the characteristics of visual perception; these enable real-time analysis. It is an open question whether similar improvements can be developed for audio residuals.

Finally, it should be noted that matching pursuit has received some attention in the image coding literature [233, 234]. With regards to this application, it may be of interest to use asymmetric atoms to improve modeling of edges in images, which is of course analogous to modeling onsets in audio signals.

### 7.2.2 Overcomplete Atomic Expansions

In addition to further work in audio coding, the developments in this thesis suggest extensions involving overcomplete signal expansions in terms of time-frequency atoms. Such issues are described in the following.

#### Evolutionary models

The sinusoidal model can be interpreted as an atomic decomposition wherein the atoms are related in an evolutionary fashion. This evolution model leads to synthesis robustness, modification capabilities, and data reduction. It would be useful to establish a similar evolution framework for atomic models based on overcomplete dictionaries.

#### Dictionary design and optimization

In matching pursuit and similar methods, the performance of the algorithm depends on the contents of the dictionary; such algorithms perform well if the dictionary contains atoms that match the signal behavior. Of course, this condition is more likely to hold for larger dictionaries, but increased dictionary size entails increased computation in the algorithm. One approach to handling this tradeoff is to generate a signal-adaptive dictionary which can be expected to perform well for a specific signal; this is only of interest, however, if such a dictionary can be arrived at by a simple heuristic analysis rather than a high-cost optimization.

Dictionary design issues in matching pursuit relate to codebook design issues for vector quantization. The primary difference is that vector quantization codebooks do not typically have the parametric structure of time-frequency dictionaries. Methods for codebook optimization are still of interest for matching pursuit, however, since the codebook adaptation can be restricted to adhere to a parametric atomic structure. This connection is briefly explored in [39]; given the extent of work that has been devoted to vector quantization techniques, further investigations of applications to time-frequency atomic models are clearly merited [216].

#### Variations of matching pursuit

Several variations of matching pursuit are described in Chapter 6; it is argued that comparison of such approaches calls for computation–rate–distortion considerations. Preliminary formalizations of such tradeoffs have appeared in the literature, but there are many open questions [224]. With computation concerns in mind, it is of interest to consider simplifications of matching pursuit. For instance, in [38], pursuit based on small subdictionaries is discussed; if the subdictionaries are well-chosen, this helps to reduce the computational requirements without substantially affecting the convergence of the atomic model. One possible way to generate useful subdictionaries is to employ a pyramid multiresolution scheme in which large scale atoms are evaluated with respect to subsampled versions of the signal; in this prospective scenario, the computation is reduced since some of the correlations are carried out in a subsampled domain.

### Refinement and modification

In Chapter 1, the application of reassignment methods to time-frequency distributions is briefly discussed. Such techniques start with a standard distribution and apply various refinements in order to achieve compaction in the time-frequency plane; this improves the readability of the distribution since the nonlinear refinements lead to enhancement of the peaks in the representation and attenuation of the cross terms [85]. In cases where the resources are available to derive a dispersed but exact overcomplete expansion using the SVD pseudo-inverse (or some other method), some form of adaptive refinement may prove useful for improving the compaction without sacrificing the accuracy of the expansion. One example of such an approach is as follows. Since the dictionary is overcomplete, some components in the expansion can be represented in terms of the other components. Then, such representation vectors can be added to the expansion while zeroing the corresponding components; in this way, the same signal reconstruction can be arrived at from a more compact model. The caveat here is that optimal compaction is still not feasible given the general complexity results presented in [39]; however, improved models may be achieved in some cases using such a method.

In addition to refinement of overcomplete expansions to improve compaction, other modifications are also of interest. In such efforts, the null space of the dictionary matrix provides a significant caveat; in short, some modifications may indeed map to this null space, meaning that a seemingly elaborate modification of the atomic components may indeed have no effect on the signal reconstruction. The open question in this area is that of establishing constraints on modifications to ensure robustness, *i.e.* predictability.

# Appendix A

# Two-Channel Filter Banks

The discrete wavelet transform is fundamentally connected to two-channel perfect reconstruction filter banks. These connections are explored in Chapter 3. Here, the relevant mathematical details involving two-channel perfect reconstruction filter banks are given.

### Two-channel critically sampled perfect reconstruction filter banks

The discrete wavelet transform can be derived in terms of critically sampled two-channel perfect reconstruction filter banks such as the one shown in Figure 3.3. The analysis of the system is carried out here in the frequency domain; the time-domain interpretation will be discussed in the next section. In terms of the $z$-transforms of the signals and filters, the output of the filter bank is:

$$\begin{aligned} \hat{X}(z) &= \frac{1}{2}\left[H_0(z)G_0(z) + H_1(z)G_1(z)\right]X(z) && \text{(A.1)} \\ &\qquad + \frac{1}{2}\left[H_0(-z)G_0(z) + H_1(-z)G_1(z)\right]X(-z) && \text{(A.2)} \\ &= T(z)X(z) \; + \; A(z)X(-z), && \text{(A.3)} \end{aligned}$$

where $T(z)$ is the direct transfer function of the filter bank and $A(z)$ characterizes the aliasing – the appearance of the modulated version $X(-z)$ in the output. The perfect reconstruction conditions are then clearly

$$\begin{aligned} T(z) &= 1 && \text{(A.4)} \\ A(z) &= 0, && \text{(A.5)} \end{aligned}$$

or, in terms of the filters,

$$\begin{aligned} G_0(z)H_0(z) \; + \; G_1(z)H_1(z) &= 2 && \text{(A.6)} \\ G_0(z)H_0(-z) \; + \; G_1(z)H_1(-z) &= 0, && \text{(A.7)} \end{aligned}$$

which can be rewritten in matrix form as

$$\begin{bmatrix} G_0(z) & G_1(z) \\ G_0(-z) & G_1(-z) \end{bmatrix} \begin{bmatrix} H_0(z) & H_0(-z) \\ H_1(z) & H_1(-z) \end{bmatrix} = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}. \tag{A.8}$$

This condition can be expressed in a shorthand form as

$$\mathbf{G}_m^T(z)\mathbf{H}_m(z) = 2\mathbf{I} \tag{A.9}$$

in terms of the *modulation* matrices $\mathbf{G}_m(z)$ and $\mathbf{H}_m(z)$ and the identity matrix $\mathbf{I}$; such modulation matrices are useful in multirate filter bank theory [2]. The design of a perfect reconstruction filter bank then amounts to the derivation of four polynomials $G_0(z), G_1(z), H_0(z)$, and $H_1(z)$ that satisfy the condition above; this issue is considered in detail in [2].

Equations (A.6) and (A.7) can be manipulated to yield a general expression relating the constituent filters; this will be especially useful for interpreting the analysis-synthesis filter bank in terms of a time-domain signal expansion. The first step in the derivation, which basically mirrors the treatment given in [2], is to rewrite Equation (A.7) as

$$G_0(z) = \frac{-G_1(z)H_1(-z)}{H_0(-z)}. \tag{A.10}$$

Substituting this expression into Equation (A.6) and solving for $G_1(z)$ yields

$$G_1(z) = \frac{-2H_0(-z)}{H_0(z)H_1(-z) - H_0(-z)H_1(z)} = \frac{-2H_0(-z)}{\det \mathbf{H}_m(z)}. \tag{A.11}$$

Similarly,

$$G_0(z) = \frac{2H_1(-z)}{\det \mathbf{H}_m(z)}. \tag{A.12}$$

Then, it is simple to establish the relationships

$$G_0(z)H_0(z) = \frac{2H_0(z)H_1(-z)}{\det \mathbf{H}_m(z)} \quad \text{and} \quad G_1(z)H_1(z) = \frac{-2H_0(-z)H_1(z)}{\det \mathbf{H}_m(z)}. \tag{A.13}$$

Noting that $\det \mathbf{H}_m(z) = -\det \mathbf{H}_m(-z)$,

$$G_1(z)H_1(z) = \frac{2H_0(-z)H_1(z)}{\det \mathbf{H}_m(-z)} = G_0(-z)H_0(-z). \tag{A.14}$$

Equation (A.6) can then be transformed into

$$G_0(z)H_0(z) + G_0(-z)H_0(-z) = 2 \tag{A.15}$$

or

$$G_1(z)H_1(z) + G_1(-z)H_1(-z) = 2. \tag{A.16}$$

Equation (A.7) can also be readily manipulated using the result of Equation (A.14). Multiplying by $H_0(z)$ yields:

$$\begin{aligned} G_0(z)H_0(z)H_0(-z) \ + \ G_1(z)H_1(-z)H_0(z) &= 0 \qquad (A.17) \\ G_1(-z)H_1(-z)H_0(-z) \ + \ G_1(z)H_1(-z)H_0(z) &= 0 \qquad (A.18) \\ \Longrightarrow \quad G_1(z)H_0(z) \ + \ G_1(-z)H_0(-z) &= 0, \qquad (A.19) \end{aligned}$$

where the last expression must hold at least where $H_1(z)$ is nonzero; indeed, no generality is actually lost here since the two-channel filter bank cannot achieve perfect reconstruction if $H_0(z)$ and $H_1(z)$ have any common zeros [2]. Similarly,

$$G_0(z)H_1(z) \ + \ G_0(-z)H_1(-z) \ = \ 0. \qquad (A.20)$$

The various $z$-transform relationships derived here for the critically sampled two-channel perfect reconstruction filter bank can be summarized in one equation:

$$G_i(z)H_j(z) \ + \ G_i(-z)H_j(-z) \ = \ 2\delta[i-j]. \qquad (A.21)$$

In the next section, this leads to an interpretation of the filter bank in terms of a biorthogonal basis.

### Perfect reconstruction and biorthogonality

By manipulating the perfect reconstruction condition in (A.21), it can be shown that a perfect reconstruction filter bank derives a signal expansion in a biorthogonal basis; the basis is related to the impulse responses of the filter bank. This relationship is of interest in that it establishes a connection between the filter bank model and the atomic model that underlie the discrete wavelet transform.

The time-domain relationship corresponding to Equation (A.21) can be derived using two properties of the $z$-transform: convolution and modulation. If

$$g[n] \overset{z}{\Longleftrightarrow} G(z) \quad \text{and} \quad h[n] \overset{z}{\Longleftrightarrow} H(z), \qquad (A.22)$$

the properties are as follows:

$$\begin{array}{ll} \text{Convolution} & \sum_k g[k]h[n-k] \ \overset{z}{\Longleftrightarrow} \ G(z)H(z) \\ \text{Modulation} & (-1)^n g[n] \ \overset{z}{\Longleftrightarrow} \ G(-z). \end{array} \qquad (A.23)$$

Using these properties to express Equation (A.21) in the time domain yields:

$$\begin{aligned} \sum_k g_i[k]h_j[m-k] \ + \ (-1)^m \sum_k g_i[k]h_j[m-k] &= 2\delta[m]\delta[i-j] \qquad (A.24) \\ \sum_k g_i[k]h_j[m-k]\,[1+(-1)^m] &= 2\delta[m]\delta[i-j]. \qquad (A.25) \end{aligned}$$

For odd $m$, the last expression simplifies trivially to $0 = 0$. For even $m$, replaced here by $2n$,

$$\sum_k g_i[k]h_j[2n-k] \;=\; \delta[n]\delta[i-j]. \tag{A.26}$$

Equivalently, the relationship can be derived as

$$\sum_k h_i[k]g_j[2n-k] \;=\; \delta[n]\delta[i-j] \tag{A.27}$$

by interchanging the filters in the convolution expression. In inner product notation, Equations (A.26) and (A.27) can be written as

$$\langle g_i[k], h_j[2n-k]\rangle \;=\; \delta[n]\delta[i-j] \tag{A.28}$$

$$\langle h_i[k], g_j[2n-k]\rangle \;=\; \delta[n]\delta[i-j], \tag{A.29}$$

respectively. The above expressions show that the impulse responses of the filters and their shifts by two, with one of the impulse responses time-reversed as indicated, constitute a biorthogonal basis for discrete-time signals (with finite energy), namely the space $l^2(z)$. Note that real filters have been implicitly assumed; for complex filters, the first terms in the inner product expressions would be conjugated. Also note that the analysis and synthesis filter banks are mathematically interchangeable; this symmetry is analogous to the equivalence of left and right matrix inverses discussed in Section 1.4.1.

The preceding derivation indicates that perfect reconstruction and biorthogonality are equivalent conditions; in the next section, this insight is used to relate filter banks and signal expansions.

### Interpretation as a signal expansion in a biorthogonal basis

Given that the impulse responses in a two-channel perfect reconstruction filter bank are related to an underlying biorthogonal basis, it is reasonable to consider the time-domain signal expansion carried out by such a filter bank. Using the notation of Figure 3.3, the channel signals are given by convolution followed by downsampling:

$$y_0[n] \;=\; \sum_m x[m]h_0[2n-m] \;=\; \langle x[m], h_0[2n-m]\rangle \tag{A.30}$$

$$y_1[n] \;=\; \sum_m x[m]h_1[2n-m] \;=\; \langle x[m], h_1[2n-m]\rangle\,. \tag{A.31}$$

Upsampling followed by convolution gives the outputs of the synthesis filters, which can be thought of as full-rate subband signals:

$$\hat{x}_0[n] \;=\; \sum_{m \text{ even}} y_0[m/2]g_0[n-m] \;=\; \sum_k y_0[k]g_0[n-2k] \;=\; \langle y_0[k], g_0[n-2k]\rangle \tag{A.32}$$

$$\hat{x}_1[n] \;=\; \sum_k y_1[k]g_1[n-2k] \;=\; \langle y_1[k], g_1[n-2k]\rangle\,. \tag{A.33}$$

The reconstructed output is thus given by

$$\begin{aligned}\hat{x}[n] &= \hat{x}_0[n] + \hat{x}_1[n] && \text{(A.34)}\\ &= \sum_k y_0[k]g_0[n-2k] + \sum_k y_1[k]g_1[n-2k] && \text{(A.35)}\\ &= \sum_k \langle x[m], h_0[2k-m]\rangle\, g_0[n-2k] + \sum_k \langle x[m], h_1[2k-m]\rangle\, g_1[n-2k] && \text{(A.36)}\\ &= \sum_{i=1}^{2}\sum_k \langle x[m], h_i[2k-m]\rangle g_i[n-2k]. && \text{(A.37)}\end{aligned}$$

Introducing the notation

$$g_{i,k}[n] = g_i[n-2k] \quad \text{and} \quad \alpha_{i,k} = \langle x[m], h_i[2k-m]\rangle, \tag{A.38}$$

the signal reconstruction can be clearly expressed as an atomic model:

$$\hat{x}[n] = \sum_{i,k} \alpha_{i,k} g_{i,k}[n]. \tag{A.39}$$

The coefficients in the atomic decomposition are derived by the analysis filter bank, and the expansion functions are time-shifts of the impulse responses of the synthesis filter bank. As noted earlier, the filter banks are interchangeable; the signal could also be written as an atomic decomposition based on the impulse responses $h_i[n]$. In any case, the atoms in the signal model correspond to the synthesis filter bank.

In this appendix, it has been shown that a critically sampled two-channel perfect reconstruction filter bank computes a signal expansion in a biorthogonal basis. Multiresolution decompositions such as the discrete wavelet transform and wavelet packets can be developed by iterating these two-channel structures. Here, it should simply be noted that the development in Equations (A.34) through (A.37) indicates the aforementioned connection between the interpretations of the wavelet transform as a filter bank model and as an atomic model; a subband signal is derived as an accumulation of weighted atoms corresponding to the impulse responses of the synthesis filter for that band. Such issues are discussed at greater length in Section 3.2.1.

# Appendix B

# Fourier Series Representations

In Chapter 5, the Fourier series is applied to a pitch-synchronous signal representation to arrive at a pitch-synchronous sinusoidal model. The details of Fourier series methods are reviewed here.

### Complex Fourier series and the discrete Fourier transform

The Fourier basis for $C^N$ is the set of harmonically related complex sinusoids:

$$\left\{ e^{j\omega_k n},\ \omega_k = \frac{2\pi k}{N} \ \text{ for } \ k = 0, 1, 2, \ldots, N-1 \right\}. \tag{B.1}$$

The complex Fourier series expansion for a signal $x[n] \in C^N$ is then

$$x[n] \;=\; \sum_{k=0}^{N-1} c_k e^{j2\pi kn/N}, \tag{B.2}$$

where the coefficients $c_k$ are given by the formulation:

$$\sum_{n=0}^{N-1} x[n] e^{-j2\pi ln/N} \;=\; \sum_{n=0}^{N-1}\sum_{k=0}^{N-1} c_k e^{j2\pi(k-l)n/N} \;=\; \sum_{k=0}^{N-1} c_k N\delta(k-l) \;=\; Nc_l \tag{B.3}$$

$$\Longrightarrow \quad c_k \;=\; \frac{1}{N}\sum_{n=0}^{N-1} x[n] e^{-j2\pi kn/N}. \tag{B.4}$$

This expression for $c_k$ is closely related to the discrete Fourier transform (DFT), which is given by the analysis and synthesis equations

$$X[k] \;=\; \sum_{n=0}^{N-1} x[n] e^{-j2\pi kn/N} \qquad \text{Analysis} \tag{B.5}$$

$$x[n] \;=\; \frac{1}{N}\sum_{k=0}^{N-1} X[k] e^{j2\pi kn/N} \qquad \text{Synthesis,} \tag{B.6}$$

where the analysis equation derives the DFT expansion coefficients or *spectrum* and the synthesis equation reconstructs the signal from those coefficients. Given the existence of fast algorithms for computing the DFT (*i.e.* the FFT), it is useful to note the simple relationship of the Fourier series coefficients and the DFT expansion:

$$c_k = \frac{X[k]}{N}. \tag{B.7}$$

### Real expansions of real signals

The Fourier expansion coefficients and the DFT spectrum are complex-valued even for real signals. For real signals, a real-valued expansion of the form

$$x[n] = \sum_{k=0}^{N-1} a_k \cos \omega_k n + b_k \sin \omega_k n \tag{B.8}$$

can be derived using Euler's equation:

$$e^{j\Theta} = \cos \Theta + j \sin \Theta. \tag{B.9}$$

For real $x[n]$,

$$x[n] = \Re\{x[n]\} = \frac{x[n] + x[n]^*}{2}. \tag{B.10}$$

Rewriting this using the complex Fourier expansion gives:

$$\begin{aligned} x[n] &= \frac{1}{2} \sum_{k=0}^{N-1} c_k e^{j\omega_k n} + c_k^* e^{-j\omega_k n} && \text{(B.11)} \\ &= \frac{1}{2} \sum_{k=0}^{N-1} c_k (\cos \omega_k n + j \sin \omega_k n) + c_k^* (\cos \omega_k n - j \sin \omega_k n) && \text{(B.12)} \\ &= \sum_{k=0}^{N-1} \left( \frac{c_k + c_k^*}{2} \right) \cos \omega_k n + j \left( \frac{c_k - c_k^*}{2} \right) \sin \omega_k n. && \text{(B.13)} \end{aligned}$$

The expansion coefficients in the Fourier cosine and sine series are thus given by:

$$\begin{aligned} a_k &= \frac{c_k + c_k^*}{2} &= \Re\{c_k\} &= \frac{\Re\{X[k]\}}{N} \\ b_k &= j\left(\frac{c_k - c_k^*}{2}\right) &= -\Im\{c_k\} &= -\frac{\Im\{X[k]\}}{N}. \end{aligned} \tag{B.14}$$

Furthermore, the spectrum of a real signal is conjugate-symmetric:

$$X[k] = X[N-k]^*, \tag{B.15}$$

which can be expressed in terms of the real and imaginary parts as

$$\begin{aligned} \Re\{X[k]\} &= \Re\{X[N-k]\} &\Longrightarrow\quad a_k &= a_{N-k} \\ \Im\{X[k]\} &= -\Im\{X[N-k]\} &\Longrightarrow\quad b_k &= -b_{N-k}. \end{aligned} \tag{B.16}$$

This underlying symmetry can be used to halve the number of coefficients needed to represent $x[n]$. For odd $N$, the simplification is:

$$\begin{aligned}
x[n] &= \sum_{k=0}^{N-1} a_k \cos\omega_k n \;+\; b_k \sin\omega_k n && \text{(B.17)}\\
&= a_0 \;+\; \sum_{k=1}^{\frac{N-1}{2}} \left[ a_k \cos\left(\frac{2\pi kn}{N}\right) \;+\; a_{N-k}\cos\left(\frac{2\pi(N-k)n}{N}\right) \right. \\
&\qquad\qquad \left. +\; b_k \sin\left(\frac{2\pi kn}{N}\right) \;+\; b_{N-k}\sin\left(\frac{2\pi(N-k)n}{N}\right)\right] && \text{(B.18)}\\
&= a_0 \;+\; \sum_{k=1}^{\frac{N-1}{2}} (a_k + a_{N-k})\cos\left(\frac{2\pi kn}{N}\right) \;+\; (b_k - b_{N-k})\sin\left(\frac{2\pi kn}{N}\right) && \text{(B.19)}\\
&= a_0 \;+\; 2\sum_{k=1}^{\frac{N-1}{2}} a_k \cos\left(\frac{2\pi kn}{N}\right) \;+\; b_k \sin\left(\frac{2\pi kn}{N}\right). && \text{(B.20)}
\end{aligned}$$

For even $N$, the result is:

$$x[n] \;=\; a_0 \;+\; a_{N/2}\cos\pi n \;+\; 2\sum_{k=1}^{\frac{N}{2}-1} a_k \cos\left(\frac{2\pi kn}{N}\right) \;+\; b_k \sin\left(\frac{2\pi kn}{N}\right). \qquad \text{(B.21)}$$

Note that in either case the $a_0$ term corresponds to the average value of the signal. Also, in the case of even $N$, the $a_{N/2}$ term corresponds to the Nyquist frequency, at which the spectrum should have zero amplitude; for the remainder, it is assumed that $a_{N/2} = 0$ for the sake of generalization.

### Magnitude-phase representations

The complex spectrum is often expressed in terms of its magnitude and phase:

$$X[k] \;=\; \Re\{X[k]\} \;+\; j\Im\{X[k]\} \;=\; |X[k]|e^{j\phi_k}, \qquad \text{(B.22)}$$

where

$$|X[k]| \;=\; \sqrt{\Re\{X[k]\}^2 \;+\; \Im\{X[k]\}^2} \quad \text{and} \quad \phi_k \;=\; \arctan\left(\frac{\Im\{X[k]\}}{\Re\{X[k]\}}\right). \qquad \text{(B.23)}$$

The magnitude-phase representation is often of interest in audio applications because the ear is relatively insensitive to phase. With this as motivation, the sine-cosine expansion of real signals discussed above can be rewritten in magnitude-phase form based on the following derivation:

$$\begin{aligned}
a\cos\Theta \;+\; b\sin\Theta &= a\left(\frac{e^{j\Theta} + e^{-j\Theta}}{2}\right) \;+\; b\left(\frac{e^{j\Theta} - e^{-j\Theta}}{2j}\right) && \text{(B.24)}\\
&= e^{j\Theta}\left(\frac{a-jb}{2}\right) \;+\; e^{-j\Theta}\left(\frac{a+jb}{2}\right) && \text{(B.25)}
\end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{2}e^{j\Theta}\sqrt{a^2+b^2}e^{-j\arctan\frac{b}{a}} + \frac{1}{2}e^{-j\Theta}\sqrt{a^2+b^2}e^{j\arctan\frac{b}{a}} \qquad \text{(B.26)}\\
&= \sqrt{a^2+b^2}\cos\left(\Theta - \arctan\frac{b}{a}\right). \qquad \text{(B.27)}
\end{aligned}
$$

Substituting $\omega_k n$ for $\Theta$, where $\omega_k = 2\pi k/N$, and incorporating a summation over $k$ yields another form for the sums in Equations (B.20) and (B.21):

$$
\begin{aligned}
x[n] &= a_0 + 2\sum_k \sqrt{a_k^2+b_k^2}\cos\left(\omega_k n - \arctan\frac{b_k}{a_k}\right) \qquad \text{(B.28)}\\
&= \frac{X[0]}{N} + \frac{2}{N}\sum_k |X[k]|\cos\left(\omega_k n + \phi_k\right), \qquad \text{(B.29)}
\end{aligned}
$$

where $|X[k]|$ and $\phi_k$ are as defined in Equation (B.23) and $k$ ranges over the half spectrum. As a check, note that:

$$X[k] = N(a_k - jb_k) = N\left(\frac{\Re\{X[k]\}}{N} - j\frac{-\Im\{X[k]\}}{N}\right) = \Re\{X[k]\} + j\Im\{X[k]\} \qquad \text{(B.30)}$$

$$\phi_k = -\arctan\frac{b_k}{a_k} = \arctan\frac{\Im\{X[k]\}}{\Re\{X[k]\}}. \qquad \text{(B.31)}$$

This magnitude-phase form is suggestive of the sinusoidal model of Chapter 2. The connection is discussed in Section 5.3, where it is shown that some of the difficulties in sinusoidal modeling can be overcome by applying the Fourier series in a pitch-synchronous manner.

# Publications

[1] M. Goodwin and M. Vetterli. Matching pursuit and signal models based on recursive filter banks. To be submitted to *IEEE Transactions on Signal Processing.*

[2] M. Goodwin and M. Vetterli. Atomic signal models based on recursive filter banks. In *Conference Record of the Thirty-First Asilomar Conference on Signals, Systems, and Computers*, November 1997.

[3] M. Goodwin and M. Vetterli. Atomic decompositions of audio signals. In *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, October 1997.

[4] M. Goodwin. Matching pursuit with damped sinusoids. In *IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, 3:2037–2040, May 1997.

[5] P. Prandoni, M. Goodwin, and M. Vetterli. Optimal segmentation for signal modeling and compression. In *IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, 3:2029–2032, May 1997.

[6] M. Goodwin. Nonuniform filter bank design for audio signal modeling. In *Conference Record of the Thirtieth Asilomar Conference on Signals, Systems, and Computers*, 2:1229–1233, November 1996.

[7] M. Goodwin and M. Vetterli. Time-frequency signal models for music analysis, transformation, and synthesis. In *Proceedings of the IEEE-SP International Symposium on Time-Frequency and Time-Scale Analysis*, pp. 133–6, June 1996.

[8] M. Goodwin. Residual modeling in music analysis-synthesis. In *IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, 2:1005–1008, May 1996.

[9] G. Chang, M. Goodwin, V. Goyal, T. Kalker. *Solutions Manual* for *Wavelets and Subband Coding* by M. Vetterli and J. Kovacevic Prentice-Hall, 1995.

[10] M. Goodwin and A. Kogon. Overlap-add synthesis of nonstationary sinusoids. In *Proceedings of the International Computer Music Conference*, pp. 355–356, September 1995.

[11] M. Goodwin and X. Rodet. Efficent Fourier synthesis of nonstationary sinusoids. In *Proceeedings of the International Computer Music Conference*, pp. 333–334, September 1994.

[12] M. Goodwin. Frequency-independent beamforming. In *Proceedings of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pp. 60-3, October 1993.

[13] M. Goodwin and G. Elko. Constant beamwidth beamforming. In *IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, 1:169–72, April 1993. Also in *IEEE Techology Update Series: Signal Processing Applications and Technology*, ed. J. Ackenhausen, pp. 499–502, 1995.

[14] G. Elko and M. Goodwin. Beam dithering: Acoustic feedback control using a modulated-directivity loudspeaker array. In *IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, 1:173–6, April 1993.

[15] M. Goodwin and G. Elko. Beam dithering: Acoustic feedback reduction using a modulated-directivity loudspeaker array. Presented at the *92nd Meeting of the Audio Engineering Society*, October 1992. Preprint 3384.

[16] G. Elko, M. Goodwin, R. Kubli, J. West. Electret Transducer Array and Fabrication Technique. AT&T Bell Labs, 1992. Patent number 5388163.

[17] M. Goodwin. Implementation and Applications of Electroacoustic Array Beamformers. S.M. Thesis, MIT, 1992.

# Bibliography

[1] L. R. Rabiner and R. W. Schafer, *Digital Processing of Speech Signals.* Englewood Cliffs, NJ: Prentice-Hall, 1978.

[2] M. Vetterli and J. Kovačević, *Wavelets and Subband Coding.* Englewood Cliffs, NJ: Prentice-Hall, 1995.

[3] K. R. Rao and P. Yip, *Discrete Cosine Transform: Algorithms, Advantages, Applications.* Boston, MA: Academic Press, 1990.

[4] H. S. Malvar, *Signal Processing With Lapped Transforms.* Boston, MA: Artech House, 1992.

[5] N. Jayant, J. Johnston, and B. Safranek, "Signal compression based on models of human perception," *Proceedings of the IEEE*, vol. 81, pp. 1385–1422, October 1993.

[6] A. Gersho, "Advances in speech and audio compression," *Proceedings of the IEEE*, vol. 82, pp. 900–918, June 1994.

[7] K. Brandenburg and G. Stoll, "ISO-MPEG-1 audio: A generic standard for coding of high-quality digital audio," *Journal of the Audio Engineering Society*, vol. 42, pp. 780–791, October 1994.

[8] D. Pan, "A tutorial on MPEG/audio compression," *IEEE Multimedia*, vol. 2, pp. 60–74, Summer 1995.

[9] C. Todd *et al.*, "AC-3: Flexible perceptual coding for audio transmission and storage," in *Proceedings of the 96th Convention of the Audio Engineering Society*, February 1994. Preprint 3796.

[10] K. Gosse *et al.*, "Subband audio coding with synthesis filters minimizing a perceptual criterion," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, pp. 347–350, April 1997.

[11] K. Gosse, O. Pothier, and P. Duhamel, "Optimizing the synthesis filter bank in audio coding for minimum distortion using a frequency weighted psychoacoustic criterion," in *Proceedings of the IEEE ASSP Workshop on Applications of Signal Processing to Audio and Acoustics*, pp. 191–194, October 1995.

[12] J. P. Princen and A. B. Bradley, "Analysis/synthesis filter bank design based on time domain aliasing cancellation," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 34, pp. 1153 – 1161, October 1986.

[13] J. Princen and J. Johnston, "Audio coding with signal adaptive filterbanks," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 5, pp. 3071–3074, May 1995.

[14] D. Sinha and A. H. Tewfik, "Low bit rate transparent audio compression using adapted wavelets," *IEEE Transactions on Signal Processing*, vol. 41, pp. 3463–3479, December 1993.

[15] D. Sinha and J. Johnston, "Audio compression at low bit rates using a signal adaptive switched filterbank," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, pp. 1053–1056, May 1996.

[16] K. Tsutsui *et al.*, "ATRAC: Adaptive transform acoustic coding for MiniDisc," in *Proceedings of the 93rd Convention of the Audio Engineering Society*, October 1992. Preprint 3456.

[17] S. Shlien, "The modulated lapped transform, its time-varying forms, and its application to audio coding standards," *IEEE Transactions on Speech and Audio Processing*, vol. 5, pp. 359–366, July 1997.

[18] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Transactions on Signal Processing*, vol. 41, pp. 3445–3462, December 1993.

[19] A. Said and W. A. Pearlman, "An image multiresolution representation for lossless and lossy compression," *IEEE Transactions on Image Processing*, vol. 5, pp. 1303–1310, September 1996.

[20] P. P. Vaidyanathan, *Multirate Systems and Filter Banks.* Englewood Cliffs, NJ: Prentice-Hall, 1993.

[21] J. O. Smith, "Physical modeling synthesis update," *Computer Music Journal*, vol. 20, pp. 44–56, Summer 1996.

[22] J. O. Smith, *Techniques for Digital Filter Design and System Identification With Application to the Violin.* PhD thesis, Stanford University, June 1983.

[23] J. Makhoul, "Linear prediction: A tutorial review," *Proceedings of the IEEE*, vol. 63, pp. 561–580, April 1975.

[24] A. Gersho, "Speech coding," in *Speech Analysis and Synthesis and Man-Machine Speech Communications for Air Operations*, pp. 3/1–3/14, May 1990.

[25] D. Griffin and J. Lim, "Multiband excitation vocoder," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 36, pp. 1223–1235, August 1988.

[26] J. A. Moorer, "The use of linear prediction of speech in computer music applications," *Journal of the Audio Engineering Society*, vol. 27, pp. 134–140, March 1979.

[27] X. Rodet, "Time-domain formant-wave-function synthesis," *Computer Music Journal*, vol. 8, pp. 9–14, Fall 1984.

[28] K. Karplus and A. Strong, "Digital synthesis of plucked-string and drum timbres," *Computer Music Journal*, vol. 7, pp. 43–55, Summer 1983.

[29] D. A. Jaffe and J. O. Smith, "Extensions of the Karplus-Strong plucked-string algorithm," *Computer Music Journal*, vol. 7, pp. 56–69, Summer 1983.

[30] G. Evangelista and S. Cavaliere, "Karplus-Strong parameter estimation," in *Proceedings of the Workshop on Physical Model Synthesis*, June 1996.

[31] J.-C. Risset and M. V. Matthews, "Analysis of musical-instrument tones," *Physics Today*, vol. 22, pp. 23–30, February 1969.

[32] M. D. Freedman, "Analysis of musical instrument tones," *Journal of the Acoustical Society of America*, vol. 41, pp. 793–806, April 1967.

[33] C. Roads, "Introduction to granular synthesis," *Computer Music Journal*, vol. 12, pp. 11–13, Summer 1988.

[34] C. Roads, *The Computer Music Tutorial.* Cambridge, MA: MIT Press, 1994.

[35] F. R. Moore, *Elements of Computer Music.* Englewood Cliffs, NJ: Prentice Hall, 1990.

[36] X. Serra and J. Smith, "Spectral modeling synthesis: A sound analysis/synthesis system based on a deterministic plus stochastic decomposition," *Computer Music Journal*, vol. 14, pp. 12–24, Winter 1990.

[37] H. Lee and K. Buckley, "Heart beat data compression using temporal beats alignment and 2-d transforms," in *Conference Record of the Thirtieth Asilomar Conference on Signals, Systems, and Computers*, vol. 2, pp. 1224–1228, November 1996.

[38] S. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Transactions on Signal Processing*, vol. 41, pp. 3397–3415, December 1993.

[39] G. Davis, *Adaptive Nonlinear Approximations.* PhD thesis, New York University, September 1994.

[40] R. Coifman and M. Wickerhauser, "Entropy-based algorithms for best basis selection," *IEEE Transactions on Information Theory*, vol. 38, pp. 713–718, March 1992.

[41] K. Ramchandran and M. Vetterli, "Best wavelet packet bases in a rate-distortion sense," *IEEE Transactions on Image Processing*, vol. 2, pp. 160–75, April 1993.

[42] S. Chen, D. Donoho, and M. Saunders, "Atomic decomposition by basis pursuit," Tech. Rep. 479, Stanford University, February 1996. Available at playfair.stanford.edu.

[43] S. Chen and D. Donoho, "Basis pursuit," in *Proceedings of the Twenty-Eighth Asilomar Conference on Signals, Systems and Computers*, vol. 1, pp. 41–44, November 1994.

[44] B. Natarajan, "Filtering random noise from deterministic signals via data compression," *IEEE Transactions on Signal Processing*, vol. 43, pp. 2595–2605, November 1995.

[45] S. F. Boll, "Suppression of acoustic noise in speech using spectral subtraction," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 27, pp. 113–120, April 1979.

[46] D. L. Donoho, "De-noising by soft-thresholding," *IEEE Transactions on Information Theory*, vol. 41, pp. 613–627, May 1995.

[47] J. Berger and C. Nichols, "Brahms at the piano: An analysis of data from the Brahms cylinder," *Leonardo Music Journal*, vol. 4, pp. 23–30, 1994.

[48] J. Berger, R. Coifman, and M. Goldberg, "Removing noise from music using local trigonometric bases and wavelet packets," *Journal of the Audio Engineering Society*, vol. 42, pp. 808–818, October 1994.

[49] S. G. Chang, B. Yu, and M. Vetterli, "Image denoising via lossy compression and wavelet thresholding," in *Proceedings of the International Conference on Image Processing*, vol. 1, pp. 604–607, October 1997.

[50] D. A. Berkeley and O. M. Mitchell, "Seeking the ideal in "hands-free" telephony," *Bell Laboratories Record*, vol. 52, pp. 318–325, November 1974.

[51] O. M. Mitchell and D. A. Berkeley, "Reduction of long-time reverberation by a center-clipping process," *Journal of the Acoustical Society of America*, vol. 47, p. 84, 1970.

[52] J. Benesty, D. R. Morgan, and M. M. Sondhi, "A better understanding and an improved solution to the problems of stereophonic acoustic echo cancellation," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, pp. 303–306, April 1997.

[53] M. M. Sonhdi, "New methods of pitch extraction," *IEEE Transactions on Audio and Electroacoustics*, vol. 16, pp. 262–266, June 1968.

[54] P. Frampton, *Frampton Comes Alive!* A & M Records, Inc., 1976.

[55] J. Chowning, "The synthesis of complex audio spectra by means of frequency modulation," *Journal of the Audio Engineering Society*, vol. 21, pp. 46–54, September 1973.

[56] W. B. Kleijn, "Encoding speech using prototype waveforms," *IEEE Transactions on Speech and Audio Processing*, vol. 1, pp. 386–399, October 1993.

[57] R. J. McAulay and T. F. Quatieri, "Speech analysis/synthesis based on a sinusoidal representation," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 34, pp. 744 – 754, August 1986.

[58] G. Strang, *Linear Algebra And Its Applications.* Harcourt Brace Jovanovich, 3rd ed., 1988.

[59] H. S. Malvar and D. H. Staelin, "The LOT: Transform coding without blocking effects," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, pp. 553 – 559, April 1989.

[60] C. Herley, J. Kovačević, K. Ramchandran, and M. Vetterli, "Tilings of the time-frequency plane: Construction of arbitrary orthogonal bases and fast tiling algorithms," *IEEE Transactions on Signal Processing*, vol. 41, pp. 3341–3359, December 1993.

[61] K. Dobson *et al.*, "A low complexity wavelet based audio compression method," in *Proceedings of the Data Compression Conference*, p. 433, March 1996.

[62] M. Goodwin and M. Vetterli, "Atomic decompositions of audio signals," in *Proceedings of the IEEE ASSP Workshop on Applications of Signal Processing to Audio and Acoustics*, October 1997.

[63] I. Daubechies, *Ten Lectures on Wavelets.* Philadelphia, PA: Society for Industrial and Applied Mathematics, 1992.

[64] Z. Czetkovic, *Overcomplete Expansions for Digital Signal Processing.* PhD thesis, University of California at Berkeley, December 1995.

[65] D. Donoho and I. Johnstone, "Ideal denoising in an orthonormal basis chosen from a library of bases," Tech. Rep. 461, Stanford University, September 1994. Available at playfair.stanford.edu.

[66] S. Mallat, D. Donoho, and A. Willsky, "Best basis algorithm for signal enhancement," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, pp. 1561–1564, May 1995.

[67] B. Rao, "Analysis and extensions of the FOCUSS algorithm," in *Conference Record of the Thirtieth Asilomar Conference on Signals, Systems, and Computers*, vol. 2, pp. 1218–1223, November 1996.

[68] J. Adler, B. Rao, and K. Kreutz-Delgado, "Comparison of basis selection methods," in *Conference Record of the Thirtieth Asilomar Conference on Signals, Systems, and Computers*, vol. 1, pp. 252–257, November 1996.

[69] B. Natarajan, "Sparse approximate solutions to linear systems," *SIAM Journal on Computing*, vol. 24, pp. 227–234, April 1995.

[70] Z. Czetkovic and M. Vetterli, "Oversampled filter banks," *IEEE Transactions on Signal Processing*, To appear.

[71] D. Gabor, "Theory of communication," *Journal of the Institution of Electrical Engineers*, vol. 93, pp. 429–457, November 1946.

[72] D. Gabor, "Acoustical quanta and the theory of hearing," *Nature*, vol. 159, pp. 591–594, May 1947.

[73] W. F. Heisenberg, *Collected Works.* W. Blum, H.-P. Dürr, and H. Rechenberg, eds. Berlin: Springer-Verlag, 1984.

[74] N. Wiener, “Spatio-temporal continuity, quantum theory, and music,” in *The Concepts of Space and Time* (M. Capek, ed.), pp. 539–546, Boston, MA: D. Reidel Publishing Company, 1975.

[75] L. Cohen, “The uncertainty principle in signal analysis,” in *Proceedings of the IEEE-SP International Symposium on Time-Frequency and Time-Scale Analysis*, pp. 182–185, October 1994.

[76] D. L. Jones and T. W. Parks, “Generation and combination of grains for music synthesis,” *Computer Music Journal*, vol. 12, pp. 27–34, Summer 1988.

[77] B. Truax, “Discovering inner complexity: Time shifting and transposition with a real-time granulation technique,” *Computer Music Journal*, vol. 18, pp. 38–48, Summer 1994.

[78] B. Truax, “Real-time granular synthesis with a digital signal processor,” *Computer Music Journal*, vol. 12, pp. 14–26, Summer 1988.

[79] S. Mallat, “A theory for multiresolution signal decomposition: The wavelet representation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, pp. 674–693, July 1989.

[80] S. Mann and S. Haykin, “’Chirplets’ and ’warblets’: Novel time-frequency methods,” *Electronics Letters*, vol. 28, pp. 114–116, January 16 1992.

[81] R. G. Baraniuk and D. L. Jones, “Shear madness: New orthonormal bases and frames using chirp functions,” *IEEE Transactions on Signal Processing*, vol. 41, pp. 3543–3549, December 1993.

[82] G. Evangelista, “The discrete-time warped frequency wavelet transform,” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, pp. 2105–2108, April 1997.

[83] L. Cohen, “Time-frequency distributions – a review,” *Proceedings of the IEEE*, vol. 77, pp. 941–981, July 1989.

[84] S. Kadambe and G. F. Boudreaux-Bartels, “A comparison of the existence of ’cross terms’ in the Wigner distribution and the squared magnitude of the wavelet transform and the short time Fourier transform,” *IEEE Transactions on Signal Processing*, vol. 40, pp. 2498–2517, October 1992.

[85] F. Auger and P. Flandrin, “Improving the readibility of time-frequency and time-scale representations by the reassignment method,” *IEEE Transactions on Signal Processing*, vol. 43, pp. 1068–1089, May 1995.

[86] W. Pielemeier and G. Wakefield, “A high resolution time-frequency representation for musical instrument signals,” *Journal of the Acoustical Society of America*, vol. 99, pp. 2382–2396, April 1996.

[87] W. Krattenthaler and F. Hlawatsch, “Time-frequency design and processing of signals via smoothed Wigner distributions,” *IEEE Transactions on Signal Processing*, vol. 41, pp. 278–287, January 1993.

[88] F. Plante, G. Meyer, and W. Ainsworth, "Speech signal analysis with reallocated spectrogram," in *Proceedings of the IEEE-SP International Symposium on Time-Frequency and Time-Scale Analysis*, pp. 640–643, October 1994.

[89] A. Bultan, "A four-parameter atomic decomposition of chirplets," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 5, pp. 3625–3628, April 1997.

[90] E. Moulines and J. Laroche, "Non-parametric techniques for pitch-scale and time-scale modification of speech," *Speech Communication*, vol. 16, pp. 175–205, February 1995.

[91] M. Goodwin and M. Vetterli, "Time-frequency signal models for music analysis, transformation, and synthesis," in *Proceedings of the IEEE-SP International Symposium on Time-Frequency and Time-Scale Analysis*, pp. 133–136, June 1996.

[92] V. Goyal, M. Vetterli, and N. Thao, "Quantized overcomplete expansions in $\Re^{\mathrm{n}}$: Analysis, synthesis and algorithms," *IEEE Transactions on Information Theory*, To appear.

[93] T. F. Quatieri and R. J. McAulay, "Speech transformations based on a sinusoidal representation," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 34, pp. 1449 – 1464, December 1986.

[94] T. F. Quatieri and R. J. McAulay, "Shape invariant time-scale and pitch modification of speech," *IEEE Transactions on Signal Processing*, vol. 40, pp. 497–510, March 1992.

[95] J. S. Marques and L. B. Almeida, "Frequency-varying sinusoidal modeling of speech," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, pp. 763–765, May 1989.

[96] L. B. Almeida and F. M. Silva, "Variable-frequency synthesis: An improved harmonic coding scheme," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, pp. 27.5.1–27.5.4, March 1984.

[97] J. S. Marques and A. J. Abrantes, "Hybrid harmonic coding of speech at low bit-rates," *Speech Communication*, vol. 14, pp. 231–247, June 1994.

[98] J. Laroche, Y. Stylianou, and E. Moulines, "HNM: A simple, efficient harmonic + noise model for speech," in *Proceedings of the IEEE ASSP Workshop on Applications of Signal Processing to Audio and Acoustics*, pp. 169–172, October 1993.

[99] J. M. Kates, "Speech enhancement based on a sinusoidal model," *Journal of Speech and Hearing Research*, vol. 37, pp. 449–464, April 1994.

[100] X. Serra, *A System for Sound Analysis/Transformation/Synthesis Based On A Deterministic Plus Stochastic Decomposition.* PhD thesis, Stanford University, October 1989.

[101] E. B. George and M. J. T. Smith, “Analysis-by-synthesis/overlap-add sinudoidal modeling applied to the analysis and synthesis of musical tones,” *Journal of the Audio Engineering Society*, vol. 40, pp. 497–516, June 1992.

[102] X. Rodet and P. Depalle, “Spectral envelopes and inverse FFT synthesis,” *Proceedings of the 93rd Convention of the Audio Engineering Society*, October 1992. Preprint 3393.

[103] K. Fitz and L. Haken, “Sinusoidal modeling and manipulation using Lemur,” *Computer Music Journal*, vol. 20, pp. 44–59, Winter 1996.

[104] T. F. Quatieri, R. B. Dunn, R. J. McAulay, and T. Hanna, “Time-scale modification of complex acoustic signals in noise,” Tech. Rep. 990, Lincoln Laboratory, M.I.T., February 1994.

[105] T. F. Quatieri, R. B. Dunn, and T. E. Hanna, “Time-scale modification with temporal envelope invariance,” in *Proceedings of the IEEE ASSP Workshop on Applications of Signal Processing to Audio and Acoustics*, pp. 127–130, October 1993.

[106] L. B. Almeida and J. M. Tribolet, “Nonstationary spectral modeling of voiced speech,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 31, pp. 664–678, June 1983.

[107] Y. Ding and X. Qian, “Estimating sinusoidal parameters of musical tones based on global waveform fitting,” in *Proceedings of the IEEE Workshop on Multimedia Signal Processing*, pp. 95–100, June 1997.

[108] D. L. Wessel, “Timbre space as a musical control structure,” *Computer Music Journal*, vol. 3, pp. 45–52, Summer 1979.

[109] K. Hamdy, M. Ali, and A. Tewfik, “Low bit rate high quality audio coding with combined harmonic and wavelet representations,” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, pp. 1045–1048, May 1996.

[110] M. Goodwin, “Residual modeling in music analysis-synthesis,” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, pp. 1005–1008, May 1996.

[111] J. Allen and L. Rabiner, “A unified approach to short-time Fourier analysis and synthesis,” *Proceedings of the IEEE*, vol. 65, pp. 1558–1564, November 1977.

[112] R. Crochiere, “A weighted overlap-add method of short-time Fourier analysis/synthesis,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 28, pp. 99–102, February 1980.

[113] J. L. Flanagan and R. M. Golden, “Phase vocoder,” *Bell System Technical Journal*, vol. 45, pp. 1493 –1509, November 1966.

[114] J. Allen, “Short term spectral analysis, synthesis, and modification by discrete Fourier transform,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 25, pp. 235–238, June 1977.

[115] M. R. Portnoff, "Time-frequency representation of digital signals and systems based on short-time Fourier analysis," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 28, pp. 55–69, February 1980.

[116] M. R. Portnoff, "Implementation of the digital phase vocoder using the fast Fourier transform," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 24, pp. 243–248, June 1976.

[117] D. Griffin and J. Lim, "Signal estimation from modified short-time Fourier transform," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 32, pp. 236–243, April 1984.

[118] M. Dolson, "The phase vocoder: A tutorial," *Computer Music Journal*, vol. 10, pp. 14–27, Winter 1986.

[119] J. A. Moorer, "The use of the phase vocoder in computer music applications," *Journal of the Audio Engineering Society*, vol. 26, pp. 42–45, January/February 1978.

[120] M. Vetterli, "Filter banks allowing perfect reconstruction," *Signal Processing*, vol. 10, pp. 219–244, April 1986.

[121] F. Léonard, "Referencing the phase to the centre of the spectral window. why?," *Mechanical Systems and Signal Processing*, vol. 2, pp. 75–90, January 1997.

[122] F. J. Harris, "On the use of windows for harmonic analysis with the discrete Fourier transform," *Proceedings of the IEEE*, vol. 66, pp. 51–83, January 1978.

[123] A. H. Nuttall, "Some windows with very good sidelobe behavior," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 29, pp. 84–91, February 1981.

[124] E. A. Lee and D. G. Messerschmitt, *Digital Communication.* Boston, MA: Kluwer Academic Publishers, 1988.

[125] H. Bölcskei and F. Hlawatsch, "Oversampled filter banks: Optimal noise shaping, design freedom, and noise analysis," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, pp. 2453–2456, April 1997.

[126] M. Dolson, *A tracking phase vocoder and its use in the analysis of ensemble sounds.* PhD thesis, University of California at San Diego, 1983.

[127] G. Oetken, T. W. Parks, and H. W. Schussler, "New results in the design of digital interpolators," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 23, pp. 301–308, June 1975.

[128] M. Slaney, D. Naar, and R. Lyon, "Auditory model inversion for sound separation," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, pp. 77–80, April 1994.

[129] M. Slaney, M. Covell, and B. Lassiter, "Automatic audio morphing," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, pp. 1001–1004, May 1996.

[130] D. Arfib and N. Delprat, "Musical transformations using the modification of time-frequency images," *Computer Music Journal*, vol. 17, pp. 66–72, Summer 1993.

[131] H. Dudley, "The vocoder," *Bell Laboratories Record*, vol. 17, pp. 122–126, 1939.

[132] A. Freed, X. Rodet, and P. Depalle, "Synthesis and control of hundreds of sinusoidal partials on a desktop computer without custom hardware," in *Proceedings of the Fourth International Conference on Signal Processing Applications and Technology*, vol. 2, pp. 1024–1030, September 1993.

[133] M. Bertocco, C. Offelli, and D. Petri, "Analysis of damped sinusoidal signals via a frequency-domain interpolation algorithm," *IEEE Transactions on Instrumentation and Measurement*, vol. 43, pp. 245–250, April 1994.

[134] P. Prandoni, M. Goodwin, and M. Vetterli, "Optimal segmentation for signal modeling and compression," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, pp. 2029–2032, April 1997.

[135] R. Kumaresan and D. Tufts, "Estimating the parameters of exponentially damped sinusoids and pole-zero modeling in noise," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 30, pp. 833–840, December 1982.

[136] D. Tufts and R. Kumaresan, "Estimation of frequencies of multiple sinusoids: Making linear prediction perform like maximum likelihood," *Proceedings of the IEEE*, vol. 70, pp. 975–989, September 1982.

[137] B. Friedlander and J. Francos, "Estimation of amplitude and phase parameters of multicomponent signals," *IEEE Transactions on Signal Processing*, vol. 43, pp. 917–925, April 1995.

[138] R. Roy and T. Kailath, "ESPRIT – Estimation of signal parameters via rotational invariance techniques," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, pp. 984–995, July 1989.

[139] S. Kay and S. Marple, "Spectrum analysis – a modern perspective," *Proceedings of the IEEE*, vol. 69, pp. 1380–1419, November 1981.

[140] D. J. Thomson, "Spectrum estimation and harmonic analysis," *Proceedings of the IEEE*, vol. 70, pp. 1055–1096, September 1982.

[141] S. Kay, *Modern Spectral Estimation: Theory and Application.* Englewood Cliffs, NJ: Prentice Hall, 1988.

[142] A. Freed, "Bring your own control to additive synthesis," in *Proceedings of the International Computer Music Conference*, pp. 303–306, September 1995.

[143] X. Xie and R. J. Evans, "Multiple target tracking and multiple frequency line tracking using hidden Markov models," *IEEE Transactions on Signal Processing*, vol. 39, pp. 2659–2676, December 1991.

[144] R. F. Barrett and D. A. Holdsworth, "Frequency tracking using hidden Markov models with amplitude and phase information," *IEEE Transactions on Signal Processing*, vol. 41, pp. 2965–2976, October 1993.

[145] P. Depalle, G. Garcia, and X. Rodet, "Tracking of partials for additive sound synthesis using hidden Markov models," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, pp. 225–228, April 1993.

[146] G. J. Adams and R. J. Evans, "Neural networks for frequency line tracking," *IEEE Transactions on Signal Processing*, vol. 42, pp. 936–941, April 1994.

[147] A. Wang, *Instantaneous and Frequency-Warped Signal Processing Techniques for Auditory Source Separation.* PhD thesis, Stanford University, August 1994.

[148] T. F. Quatieri and R. J. McAulay, "Phase modelling and its application to sinusoidal transform coding," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, pp. 1713–1715, April 1986.

[149] R. J. McAulay and T. F. Quatieri, "Magnitude-only reconstruction using a sinusoidal speech model," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, pp. 27.6.1–27.6.4, March 1984.

[150] R. J. McAulay and T. F. Quatieri, "Computationally efficient sine-wave synthesis and its application to sinusoidal transform coding," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, pp. 370–373, April 1988.

[151] M. Tabei and M. Ueda, "FFT multi-frequency synthesizer," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, pp. 1431–1434, April 1988.

[152] M. Goodwin and X. Rodet, "Efficient Fourier synthesis of nonstationary sinusoids," in *Proceedings of the International Computer Music Conference*, pp. 333–334, September 1994.

[153] M. Goodwin and A. Kogon, "Overlap-add synthesis of nonstationary sinusoids," in *Proceedings of the International Computer Music Conference*, pp. 355–356, September 1995.

[154] E. B. George and M. J. T. Smith, "Speech analysis/synthesis and modification using an analysis-by-synthesis/overlap-add sinusoidal model," *IEEE Transactions on Speech and Audio Processing*, vol. 5, pp. 389–406, September 1997.

[155] T. F. Quatieri and R. J. McAulay, "Peak-to-RMS reduction of speech based on a sinusoidal model," *IEEE Transactions on Signal Processing*, vol. 39, pp. 273–288, February 1991.

[156] J. Laroche and M. Dolson, "About this phasiness business," in *Proceedings of the IEEE ASSP Workshop on Applications of Signal Processing to Audio and Acoustics*, October 1997.

[157] D. Wessel, June 1996. Personal communication.

[158] G. Wolberg, "Recent advances in image morphing," in *Proceedings of Computer Graphics International*, pp. 64–71, June 1996.

[159] J. S. Marques and L. B. Almeida, "New basis functions for sinusoidal decompositions," in *Proceedings of the European Conference on Electrotechnics*, pp. 48–51, June 1988.

[160] J. S. Marques and L. B. Almeida, "Sinusoidal modeling of speech: Representation of unvoiced sounds with narrow-band basis functions," in *Proceedings of the European Signal Processing Conference*, vol. 2, pp. 891–894, September 1988.

[161] C. van den Branden Lambrecht and M. Karrakchou, "Wavelet packets-based high-resolution spectral estimation," *Signal Processing*, vol. 47, pp. 135–144, November 1995.

[162] B. Moore, *An Introduction to the Psychology of Hearing.* San Diego, CA: Academic Press, 1997.

[163] J. Benedetto and A. Teolis, "An auditory motivated time-scale signal representation," in *Proceedings of the IEEE-SP International Symposium on Time-Frequency and Time-Scale Analysis*, pp. 49–52, October 1992.

[164] M. Unser and A. Aldroubi, "A review of wavelets in biomedical applications," *Proceedings of the IEEE*, vol. 84, pp. 626–638, April 1996.

[165] P. J. Burt and E. H. Adelson, "The Laplacian pyramid as a compact image code," *IEEE Transactions on Communications*, vol. 31, pp. 532–540, April 1983.

[166] N. Fliege and U. Zölzer, "Multi-complementary filter bank," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, pp. 193–196, April 1993.

[167] M. Schönle, N. Fliege, and U. Zölzer, "Parametric approximation of room impulse responses based on wavelet decomposition," in *Proceedings of the IEEE ASSP Workshop on Applications of Signal Processing to Audio and Acoustics*, pp. 68–71, October 1993.

[168] S. Levine, T. Verma, and J. Smith, "Alias-free, multiresolution sinusoidal modeling for polyphonic wideband audio," in *Proceedings of the IEEE ASSP Workshop on Applications of Signal Processing to Audio and Acoustics*, October 1997.

[169] M. Rodriguez-Hernandez and F. Casajus-Quiros, "Improving time-scale modification of audio signals using wavelets," in *Proceedings of the Fifth International Conference on Signal Processing Applications and Technology*, vol. 2, pp. 1573–1577, October 1994.

[170] D. Anderson, "Speech analysis and coding using a multi-resolution sinusoidal transform," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, pp. 1037–1040, May 1996.

[171] D. Ellis and B. Vercoe, "A wavelet-based sinusoid model of sound for auditory signal separation," in *Proceedings of the International Computer Music Conference*, pp. 86–89, 1991.

[172] R. E. Bellman, *Dynamic Programming.* Princeton, NJ: Princeton University Press, 1957.

[173] L. R. Rabiner and B.-H. Juang, *Fundamentals of Speech Recognition.* Englewood Cliffs, NJ: Prentice Hall, 1993.

[174] G. White, "Dynamic programming, the viterbi algorithm, and low cost speech recognition," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 413–417, April 1978.

[175] D. Bertsimas and J. Tsitsiklis, *Introduction to Linear Optimization.* Belmont, MA: Athena Scientific, 1997.

[176] Z. Xiong, C. Herley, K. Ramchandran, and M. Orchard, "Flexible time segmentations for time-varying wavelet packets," in *Proceedings of the IEEE-SP International Symposium on Time-Frequency and Time-Scale Analysis*, pp. 9–12, October 1994.

[177] K. Ramchandran, Z. Xiong, K. Asai, and M. Vetterli, "Adaptive transforms for image coding using spatially varying wavelet packets," *IEEE Transactions on Image Processing*, vol. 5, pp. 1197–1204, July 1996.

[178] M. Goodwin, "Nonuniform filter bank design for audio signal modeling," in *Conference Record of the Thirtieth Asilomar Conference on Signals, Systems, and Computers*, vol. 2, pp. 1229–1233, November 1996.

[179] S. Kwon and A. Goldberg, "An enhanced LPC vocoder with no voiced/unvoiced switch," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 32, pp. 851–858, August 1984.

[180] T. Verma, S. Levine, and J. Smith, "TMS: Transient modeling synthesis," in *Proceedings of the International Computer Music Conference*, September 1997.

[181] R. Neff, A. Zakhor, and M. Vetterli, "Very low bit rate video coding using matching pursuits," in *Proceedings of the SPIE Conference on Visual Communication and Image Processing*, vol. 2308, pp. 47–60, September 1994.

[182] H. Helmholtz, *On the Sensations of Tone as a Physiological Basis for the Theory of Music.* London: Longmans, Green, and Co., 1875.

[183] H. Fletcher, "Physical measurements of audition and their bearing on the theory of hearing," *Bell System Technical Journal*, vol. 2, pp. 145–180, October 1923.

[184] H. Fletcher and W. A. Munson, "Loudness, its definition, measurement, and calculation," *Bell System Technical Journal*, vol. 12, pp. 377–430, October 1933.

[185] E. Zwicker and E. Terhardt, "Analytical expressions for critical-band rate and critical bandwidth as a function of frequency," *Journal of the Acoustical Society of America*, vol. 68, pp. 1523–1525, November 1980.

[186] B. Moore and B. Glasberg, "Suggested formulae for calculating auditory-filter bandwidths and excitation patterns," *Journal of the Acoustical Society of America*, vol. 74, pp. 750–753, September 1983.

[187] A. Papoulis, *Probability, Random Variables, and Stochastic Processes.* New York: McGraw-Hill, Inc., 1991.

[188] J. Princen, "The design of nonuniform modulated filter banks," in *Proceedings of the IEEE-SP International Symposium on Time-Frequency and Time-Scale Analysis*, pp. 112–115, October 1994.

[189] J. Princen, "The design of nonuniform modulated filterbanks," *IEEE Transactions on Signal Processing*, vol. 43, pp. 2550–2560, November 1995.

[190] K. Nayebi, T. P. Barnwell, and M. J. T. Smith, "The design of perfect reconstruction nonuniform band filter banks," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, pp. 1781–1784, May 1991.

[191] J. Kovačević and M. Vetterli, "Perfect reconstruction filter banks with rational sampling rate changes," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, pp. 1785–1788, May 1991.

[192] J. Li, T. Nguyen, and S. Tantaratana, "A simple design method for nonuniform multirate filter banks," in *Conference Record of the Twenty-Ninth Asilomar Conference on Signals, Systems, and Computers*, vol. 2, pp. 1015–1019, November 1994.

[193] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing.* Englewood Cliffs, NJ: Prentice-Hall, 1989.

[194] J. O. Smith, July 1994. Personal communication.

[195] G. Evangelista, "Pitch-synchronous wavelet representations of speech and music signals," *IEEE Transactions on Signal Processing*, vol. 41, pp. 3313–3330, December 1993.

[196] W. Hess, *Pitch Determination of Speech Signals.* New York, NY: Springer-Verlag, 1983.

[197] Y. Medan, E. Yair, and D. Chazan, "Super resolution pitch determination of speech signals," *IEEE Transactions on Signal Processing*, vol. 39, pp. 40–48, January 1991.

[198] C. Wendt and A. Petropulu, "Pitch determination and speech segmentation using the discrete wavelet transform," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, pp. 45–48, May 1996.

[199] S. Mallat and S. Zhong, "Characterization of signals from multiscale edges," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, pp. 710–732, July 1992.

[200] J. O. Smith and P. Gossett, “A flexible sample-rate conversion method,” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, pp. 19.4.1–19.4.4, March 1984.

[201] J. Laroche, July 1996. Personal communication.

[202] F. F. Lee, “Time compression and expansion of speech by the sampling method,” *Journal of the Audio Engineering Society*, vol. 20, pp. 738–742, November 1972.

[203] M. Asi and B. Saleh, “Filter-bank approach to time scaling of speech,” in *Proceedings of the European Signal Processing Conference*, vol. 2, pp. 1343–1346, September 1990.

[204] T. F. Quatieri and R. J. McAulay, “Phase coherence in speech reconstruction for enhancement and coding applications,” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, pp. 207–210, May 1989.

[205] G. Evangelista, “Comb and multiplexed wavelet transforms and their applications to signal processing,” *IEEE Transactions on Signal Processing*, vol. 42, pp. 292–303, February 1994.

[206] G. Evangelista, “The coding gain of multiplexed wavelet transforms,” *IEEE Transactions on Signal Processing*, vol. 44, pp. 1681–1692, July 1996.

[207] A. Bregman, *Auditory Scene Analysis: The Perceptual Organization of Sound.* Cambridge, MA: MIT Press, 1990.

[208] S. Jalaleddine *et al.*, “ECG data compression techniques – a unified approach,” *IEEE Transactions on Biomedical Engineering*, vol. 37, pp. 329–343, April 1990.

[209] J. Lipeikiene, “Data compression methods. application to ECG,” *Informatica*, vol. 4, no. 1-2, pp. 57–80, 1993.

[210] L. Baghai-Ravary, S. Beet, and M. Tokhi, “The two-dimensional discrete cosine transform applied to speech data,” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, pp. 244–247, May 1996.

[211] S. Qian and D. Chen, “Signal representation using adaptive normalized Gaussian functions,” *Signal Processing*, vol. 36, pp. 1–11, March 1994.

[212] Y. Pati, R. Rezaiifar, and P. Krishnaprasad, “Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition,” in *Conference Record of the Twenty-Seventh Asilomar Conference on Signals, Systems, and Computers*, vol. 1, pp. 40–44, October 1993.

[213] S. Jaggi *et al.*, “High resolution pursuit for feature extraction,” Tech. Rep. LIDS-P-2371, MIT, November 1996.

[214] H. Feichtinger, A. Turk, and T. Strohmer, “Hierarchical parallel matching pursuit,” in *Proceedings of the SPIE – The International Society for Optical Engineering*, vol. 2302, pp. 222–232, July 1994.

[215] M. Goodwin, "Matching pursuit with damped sinusoids," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, pp. 2037–2040, April 1997.

[216] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression.* Boston, MA: Kluwer Academic Publishers, 1992.

[217] P. Huber, "Projection pursuit," *Annals of Statistics*, vol. 13, no. 2, pp. 435–475, 1985.

[218] J. Friedman and W. Stuetzle, "Projection pursuit regression," *Journal of the American Statistical Assocation*, vol. 76, pp. 817–823, 1981.

[219] G. Golub and C. V. Loan, *Matrix Computations.* Baltimore, MD: Johns Hopkins Press, 1983.

[220] R. Rezaiifar and H. Jafarkhani, "Wavelet based speech coding using orthogonal matching pursuit," in *Proceedings of the Twenty-Ninth Annual Conference on Information Sciences and Systems*, pp. 88–92, March 1995.

[221] R. A. DeVore and V. N. Temlyakov, "Some remarks on greedy algorithms," *Advances in Computational Mathematics*, vol. 5, no. 2-3, pp. 173–187, 1996.

[222] R. A. DeVore and V. N. Temlyakov, "Nonlinear approximation by trigonometric sums," *The Journal of Fourier Analysis and Applications*, vol. 2, no. 1, pp. 173–187, 1995.

[223] Z. Landau, December 1996. Personal communication.

[224] V. Goyal and M. Vetterli, "Computation-distortion characteristics of block transform coding," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 4, pp. 2729–2732, April 1997.

[225] R. Gribonval *et al.*, "Analysis of sound signals with high resolution matching pursuit," in *Proceedings of the IEEE-SP International Symposium on Time-Frequency and Time-Scale Analysis*, pp. 125–128, June 1996.

[226] B. Friedlander and A. Zeira, "Oversampled Gabor representation for transient signals," *IEEE Transactions on Signal Processing*, vol. 43, pp. 2088–2094, September 1995.

[227] C. Herley and M. Vetterli, "Wavelets and recursive filter banks," *IEEE Transactions on Signal Processing*, vol. 41, pp. 2536–2556, August 1993.

[228] S. Tomažic, "On short-time Fourier transform with single-sided exponential window," *Signal Processing*, vol. 55, pp. 141–148, December 1996.

[229] M. Unser, "Recursion in short-time signal analysis," *Signal Processing*, vol. 5, pp. 229–240, May 1983.

[230] I. Trancoso *et al.*, "Quantization issues in harmonic coders," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, pp. 382–385, April 1988.

[231] S. Levine, October 1997. Personal communication.

[232] S. Ahmadi, "New techniques for sinusoidal coding of speech at 2400 bps," in *Conference Record of the Thirtieth Asilomar Conference on Signals, Systems, and Computers*, vol. 1, pp. 770–774, November 1996.

[233] F. Bergeaud and S. Mallat, "Matching pursuit of images," in *Proceedings of the International Conference on Image Processing*, vol. 1, pp. 53–56, October 1995.

[234] S. Safavian *et al.*, "Projection pursuit image compression with variable block size segmentation," *IEEE Signal Processing Letters*, vol. 4, pp. 117–120, May 1997.