Categorization and Modeling of Sound Sources for Sound Analysis/Synthesis

Jung Suk Lee

Music Technology Area, Department of Music Research
Schulich School of Music
McGill University
Montreal, Quebec, Canada

April 2013

A dissertation submitted to McGill University in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Music.

Abstract

In this thesis, various sound analysis/re-synthesis schemes are investigated in a source/filter model framework, with emphasis on the source component. This research provides improved methods and tools for sound designers, composers and musicians to flexibly analyze and synthesize sounds used for gaming, film or computer music, ranging from abstract, complex sounds to those of real musical instruments.

First, an analysis-synthesis scheme for the reproduction of a rolling ball sound is presented. The proposed scheme is based on the assumption that the rolling sound is generated by a concatenation of micro-contacts between a ball and a surface, each having associated resonances. Contact timing information is extracted from the rolling sound using an onset detection process, allowing for segmentation of a rolling sound. Segmented sound snippets are presumed to correspond to micro-contacts between a ball and a surface; thus, subband-based linear predictions (LP) are performed to model time-varying resonances and anti-resonances. The segments are then resynthesized and overlap-added to form a complete rolling sound. A "granular" analysis/synthesis approach is also applied to various kinds of environmental sounds (rain, fireworks, walking, clapping) as an additional investigation into how the source type influences the strategic choices for the analysis/synthesis of sounds. The proposed granular analysis/synthesis system allows for flexible analysis of complex sounds and re-synthesis with temporal modification. Lastly, a novel approach to extract a pluck excitation from a recorded plucked string sound is proposed within a source / filter context using physical models. A time domain windowing method and an inverse filtering-based method are devised based on the behavior of wave propagation on the string. In addition, a parametric model of the

pluck excitation as well as a method to estimate its parameters are addressed.

Sommaire

Dans cette thèse, nous avons étudié plusieurs schémas d'analyse/synthèse dans le cadre des modèles source/filtre, avec un attention particulière portée sur la composante de source. Cette recherche améliore les méthodes ainsi que les outils fournis créateurs de sons, compositeurs et musiciens désirant analyser et synthétiser avec flexibilité des sons destinés aux jeux vidéos, au cinéma ou à la musique par ordinateur. Ces sons peuvent aller de sons abstraits et complexes à ceux provenant d'instruments de musique existants.

En premier lieu, un schéma d'analyse-synthèse est introduit permettant la reproduction du son d'une balle en train de rouler. Ce schéma est fondé sur l'hypothèse que le son de ce roulement est généré par la concaténation de micro-contacts entre balle et surface, chacune d'elles possédant sa propre série de résonances. L'information relative aux temps de contact est extraite du son du roulement que l'on cherche à reproduire au moyen d'une procédure détectant le début du son afin de le segmenter. Les segments de son ainsi isolés sont supposés correspondre aux micro-contacts entre la balle et la surface. Ainsi un algorithme de prédiction linéaire est effectué par sous-bande, préalablement extraites afin de modéliser des résonances et des anti-résonances variants dans le temps. Les segments sont ensuite re-synthétisés, superposés et additionnés pour reproduire le son du roulement dans son entier. Cette approche d'analyse/synthèse "granulaire" est également appliquée à plusieurs sons de types environnementaux (pluie, feux d'artifice, marche, claquement) afin d'explorer plus avant l'influence du type de la source sur l'analyse/synthèse des sons. Le système proposé permet une analyse flexible de sons complexes et leur synthèse, avec la possibilité d'ajouter des modifications temporelles. Enfin, une approche novatrice pour extraire le signal d'excitation d'un son de

corde pincée est présentée dans le contexte de schémas source/filtre sur une modèlisation physique. A cet effet, nous introduisons une méthode de type fenêtrage, et une méthode de filtrage inverse fondée sur le type de propagation selon laquelle l'onde se déplace le long de la corde. De plus, un modèle paramétrique de l'excitation par pincement ainsi qu'une méthode d'estimation de ces paramètres sont détaillés.

Acknowledgements

First, I would like to thank my advisors, Prof. Gary P. Scavone and Prof. Philippe Depalle, for the guidance they have given me and for supporting me. They were always there for me and had my best interests at heart. They always led me in the right direction when I was lost and supported me when I was on the right track. They have not only been my mentor but also good friends to me. I would like to thank friends at Music Tech for making my time spent in the lab cheerful and fun. In particular, folks at SPCL and CAML: Bertrand Scherrer, Corey Kereliuk, Marlon Schumacher, Brian Hamilton, Mark Zadel, Antoine Lefebvre, Vincent Freour, Shi Yong, Harry Saitis, Adam Jenkins, Aanchan Mohan. Special thanks to my Korean colleagues, Songhui Chon, Moonseok Kim, for everything. I also have to especially thank François Germain for his language help.

Also, I would like to thank my parents for their endless love and support. I am deeply indebted to them for the sacrifices that they made. I want to thank my sister Jooyeon for being the best sibling.

For the last but not all the least, I am deeply grateful to my amazing wife Minkyeong for her unending love, support, and encouragement.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The development of computer and electronic technologies has led to considerable changes in every aspect of our lives. Musical art is no exception, and these developments are enabling new creative possibilities in the fields of sound production and music composition. New technologies are not only broadening the horizon of conventional music based on acoustic instruments but also allowing for a new genre of music based completely on sounds created by electronic means.

In this respect, sound analysis/synthesis can be regarded as one of the main aspects of music technology. Sound analysis/synthesis technologies allow for transformation of sounds of existing musical instruments, as well as non-musical sounds, such as found in nature, and also enables the creation of sounds that are unlike any previously known. Therefore, many analysis/synthesis methods have been studied and developed for music composition and applications in multimedia by composers and sound designers. One of the most widely used and actively studied approaches is the analysis/synthesis method referred to as the "source/filter model." The "sound source" can refer to the energy initiating a vibration of a resonant object or a microscopic

sound element that, when combined with many other such microscopic elements, constitutes a perceptually meaningful macroscopic sound. In this context, analysis of the sound source involves a process that removes the resonant aspect from the overall sound to reveal only the initial input or driving energy, or a process that segments the sound into microscopic pieces so as to identify sound sources as basic elements that collectively constitute an overall sound. Sometimes, these processes are applied together.

In the source-filter model, the input energy, or excitation, is referred to as the "source" and provides energy to the resonant system, which is referred to as the "filter." Sources can be generally categorized into two types according to their characteristics: short, percussive-like excitations (such as the striking of a drumstick) or those that are more steady and sustained (such as a glottal pulse train for speech synthesis). There has been a large body of research conducted on analysis/re-synthesis of sounds within the context of the source/filter model, especially with respect to speech and contact-based environmental sounds. For example, in [4][5], analysis/re-synthesis schemes for the sound of human footsteps and applause are investigated. As those sounds obviously consist of isolated contact events, the sounds are first decomposed into segments representing single events, and each single event is further analyzed using a source-filter approach. Sounds generated by rolling objects involve more complex structural interactions, though various source-filter approaches have been investigated [6][7][8][9].

The granular analysis/synthesis approach involves the segmentation of an existing sound and the recombination of the resulting segments to produce a new or modified sound, which allows for analyzing the sound source as the microscopic sound element. We consider extensions to the grain remixing stage

that are derived in part from source/filter techniques. In [10], the authors propose a method to synthesize a wide variety of sounds by concatenating "sound units" chosen from a large database of sound "atoms" in such a way that the "unit descriptor" is best matched with the given target. In [11], Picard *et al.* proposed a granular analysis/synthesis technique that operates in conjunction with a physics engine governing the behavior of objects that are involved in sound generation.

Source-filter based analysis/re-synthesis techniques have also been widely used for the analysis/re-synthesis of musical instruments sounds, since most musical instruments have a design that fits this excitor/resonator paradigm. Laroche and Meillier [12] proposed a physically-inspired method of analyzing and synthesizing sources as excitation for musical instruments, particularly the piano as an exemplary instrument. In [13][14][15], a pluck excitation is extracted by inverse-filtering a given plucked string sound with the string model. Lee *et al.* [16] present a way to extract an excitation for the acoustic guitar in a non-parametric way where harmonic peaks in the short time spectrum are smoothed.

The goal of this thesis is to investigate the role of the sound source as a key element in analysis and synthesis of various kinds of sounds. To that end, we investigate three different sound analysis/re-synthesis contexts that properly take into account the type of the source involved in the sound:

1. complex interactions involving a sustained source exemplified by rolling objects,

2. extensions to granular analysis/synthesis,

3. the source extraction from a plucked string instrument, involving a short, percussive source.

A common element in this analysis of sustained sources involves segmentation. That is, the sounds must first be analyzed and separated into individual sound-producing "events" or collisions. This task becomes more complicated when the sounds produced by each event overlap, examples of which include individual water drop sounds in rain or micro-collisions between a rolling object and a surface. We introduce several novel methods to address this issue applied to contexts 1 and 2. After the sounds have been properly segmented, their resonant properties must be characterized and subsequently inverse filtered from the audio segments. This process can be pursued in different ways, depending on whether the resonances are expected to vary in time. Contexts 1 and 2 make use of variations on linear prediction to extract resonant properties of the systems, while context 3 makes use of a physical model of a plucked string. The overall goal is to extract source/filter features of the sounds to enable flexible re-synthesis.

In Chapter 2, an analysis-synthesis scheme targeting rolling sounds based on the source-filter model is proposed. For analysis, it is assumed that sounds generated by rolling objects involve a slow time-variation of the resonant properties of the surface because the micro-collisions happen at different locations as the ball moves. Thus, a given rolling sound is segmented into micro sound events, and both the resonance and anti-resonance aspects of each micro sound event are analyzed. For resynthesis, each micro sound event is synthesized by implementing filters representing the resonances and anti-resonances, and synthesized sound events are concatenated in accordance with the timing information obtained from segmentation task. Further flexibility is provided by allowing both temporal and spectral aspects of rolling segments to be varied. Thus, with the proposed system [17], sounds from various rolling situations

can be synthesized in a physically meaningful way.

In Chapter 3, in order to investigate how the source type affects the analysis/synthesis of sound from another perspective, a novel granular analysis/synthesis system specific to environmental sounds is proposed. In the grain analysis stage, transient events are detected so as to segment a given sound into grains and store them in a dictionary. In order to adapt to the characteristics of given environmental sounds, a component that can distinguish stationary/non-stationary parts in the given sound is included in the analysis stage, thus providing an opportunity to redefine the source type. Furthermore, audio features are extracted from all grains for synthesis. With the proposed granular synthesis scheme, grains are effectively modified and assembled to create flexible environmental sounds. Flexible time modification, not only at the grain level, but also at the level of a whole sound event is possible. We can synthesize variants of a given target sound by time-scaling (stretching/shrinking) and time shuffling of grains.

In Chapter 4, the last context investigated concerns a novel approach to extract the excitation from a recording of a plucked guitar string. A physical model of a string is first derived and then converted to a source-filter form. Both an inverse filtering-based source-filter decomposition and a time-windowing approach [18], which is similar to segmenting a sound with the granular approach, can be applied to obtain a pluck excitation. By taking the plucking position and the signal pickup position into account, the extracted excitation becomes compact in time, which preserves the temporal information of the finger/plectrum and string interaction [18]. In order to parametrically model extracted pluck excitations, the Liljencrants-Fant (LF) model is employed and the extended Kalman filter (EKF) is used to estimate LF model

parameters. We also discuss how to integrate the proposed extraction techniques with the physical model of the finger/plectrum introduced in [19].

Chapter 2

Analysis/Synthesis of Rolling Sounds Using a Source-Filter Approach

2.1 Introduction

In this chapter, we propose an analysis-synthesis scheme for the reproduction of a rolling ball sound. The approach is based on the assumption that the rolling sound is generated by a concatenation of micro-impacts between a ball and a surface, each having associated resonances. Contact timing information is first extracted from the rolling sound using an onset detection process. The resulting individual contact segments are subband filtered before being analyzed using linear prediction (LP) and notch filter parameter estimation. The segments are then resynthesized and overlap-added to form a complete rolling sound.

The proposed scheme can be viewed in the framework of a source/filter model. Each segment assumed to represent a micro-impact is decomposed

into source and filter components through LP analyses, allowing for modeling anti-resonances as well as resonances occuring from the ball-surface interaction. This consequently yields time-varying filters depending on the rolling trajectory which accordingly excites the modes of the surface. As both resonances and anti-resonances associated with surface modes are essential perceptual attributes of rolling sounds, the proposed scheme will contribute to flexible synthesis of various rolling sounds.

2.2 Background

The synthesis of rolling sounds has applications in virtual reality and the game industry, where high-quality parametric models of environmental sounds are important in creating a realistic and natural result. Methods for the synthesis of rolling sounds have been studied by several researchers. Van den Doel [20] proposed a source-filter approach to produce various types of sounds based on the contact-based interaction between objects. Conducting modal analyses of the objects involved in the interaction leads to a bank of resonators, and this bank is driven by a contact force empirically modeled by colored noise in a way that the physical attributes specifically associated with rolling interaction, such as the surface asperity, the ball-surface feedback and the ball size, are taken into account. In [7] and [8], a real-time physically based parametric 'cartoonification' model of a rolling object was proposed, where the high-level impact interaction model introduced by Hunt and Crossley [21] was combined with a modal synthesis technique to describe the vibrations of objects. The roughness of the surface is modeled in a simplified fashion using random values to account for the trajectory of the ball on the surface at the microscopic level so as to inform the impact interaction model to properly excite modes of

the surface. In addition, by considering the trajectory of the rolling sphere's center of mass, the proposed method could take the shape of the sphere into account. Due to the nature of simplified components, the computational load is moderate enough to enable real time synthesis. Lagrange *et al.* [9] assumed a similar ball-surface interaction but took a different approach for the analysis and resynthesis. They focused on the time domain property of the rolling ball sound as generated by many discrete contacts between the ball and the plate, rather than continuous contacts as modeled in [20]. The authors first analyzed the modes of the given rolling ball sound using the high resolution (HR) method in order to estimate a fixed resonant filter characteristics and then estimated the source signal as a series of impulses convolved with parameterized impact windows in an iterative way. However, though the method proposed by Lagrange *et al.* is in the form of source filter modeling, this method does not address the position-dependent modal property of the plate.

Another time-domain approach is proposed by Stoelinga and Chaigne [22] in which, in contrast to the approaches in the research mentioned above, a physical modeling technique is employed. The study extends the physical model of the impact between a ball and a damped plate to model a rolling ball. For the single impact, the vibration of a damped plate is modeled on the Kirchhoff-Love approximation under the appropriate assumption. The vibration is described by the wave equation, which involves the model of rigidities and the excitation force. The excitation force, caused by the interaction between the ball and the plate, is modeled by the Hertz's law of contact [23]. This single impact model is applied to account for the rolling interaction. The surface profile, which represents the roughness of the surface, is generated from random numbers and then assigned to spatial grids on which the numerical

calculation of the plate displacements is conducted. The proposed physical model is validated first by carrying out the simulation. The restitution coefficient obtained from the simulation agrees well with the measured values given a specific condition. Moreover, the model could handle the rebounds which occur when a ball rolls faster than the critical speed on a wavy plate. The Doppler effect is also revealed as expected. The result of simulation also shows that simulated rolling sounds can handle various rolling scenarios referred to as amplitude-modulation, periodic bouncing, chaotic bouncing and continuous contact. Both temporal and spectral characteristics of the simulated sounds appear to reproduce those of measured sounds to good quality, resulting in auditory resemblance. However, with this model, a high sampling rate is required to solve the wave equation using high-order finite different schemes and minimize inaccuracy of resulting mode frequencies. Also, the given physical model, which is based on the thin plate theory, is not appropriate for a thick plate.

2.3 Proposed Approach

In this chapter, it is assumed that the rolling sound is composed of a collection of micro-collisions between a rolling object and an uneven surface. The goal is to investigate the extent to which an efficient source-filter approach could be used to achieve a good synthesis quality for rolling sounds, on the basis of analysis of recorded sounds. First, a contact event estimation is performed and the sound is segmented accordingly. Then, a separate filter characteristic for each segment is estimated. In this way, it is possible to account for the varying modal property with respect to the locations of the contacts along the trajectory of the object on the surface, which in turn allows a physically

intuitive analysis. A description is first given on how to decompose the rolling sound signal into individual contact segments. Secondly, the analysis and synthesis of each segment are discussed. A filter bank splits each segment into subbands with different frequency bandwidths, which enables a better linear prediction (LP) estimation, especially for strong low-frequency resonant modes. A notch filter estimation to account for effects related to position-dependent excitation of the modes of the plate is also performed.

2.4 Detection of Contact Timings and the Definition of One Contact Sound

The present approach is based on the underlying assumption that the rolling sound results from numerous micro-collisions of a rolling object over an uneven surface. Thus, we must find the contact timings so that analyses of individual contact events can be accomplished. To this end, high-pass filtering is first performed on the signal to help distinguish contacts by their high-frequency transients. Fig. 2.1 shows a rolling sound signal, denoted as $y(n)$, its high-pass filtered version (cutoff frequency is 10kHz with sampling frequency 44.1kHz), $y_{hp}(n)$, and the spectrogram of $y(n)$. $y(n)$ was recorded using an accelerometer attached on one end of the plate.

A linear-phase high-pass filter is used to obtain $y_{hp}(n)$ so that group delays can be easily aligned. In order to detect contact timings more accurately, an onset detection process is performed on $y_{hp}(n)$.

$$E(n) = \frac{1}{N} \sum_{m=-\frac{N}{2}}^{\frac{N}{2}-1} [y_{hp}(n+m)]^2 \tag{2.1}$$

An envelope function $E(n)$ is defined as in Eq. 2.1 [24] and the *box function*

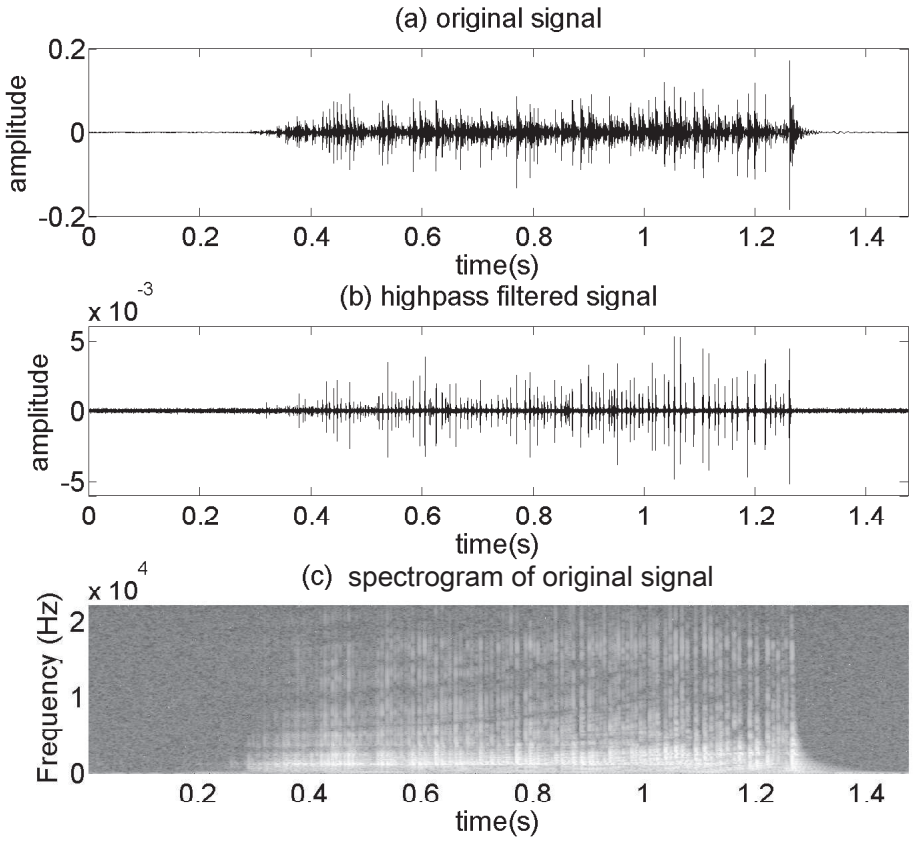

Fig. 2.1 Top: Original rolling sound $y(n)$. Middle: High-pass filtered rolling sound $y_{hp}(n)$. Bottom: Spectrogram of $y(n)$.

$b(n)$ is defined, as in Eq. 2.2, by replacing all values of $E(n)$ greater than any given threshold with a positive value α.

$$b(n) = \begin{cases} \alpha & \text{if} \quad E(n) > \text{threshold} \\ 0 & \text{otherwise.} \end{cases} \tag{2.2}$$

Fig. 2.2(a) shows the $E(n)$ of $y(n)$ and Fig. 2.2(b) shows an enlarged portion of $E(n)$ and its associated $b(n)$. In order to detect the onset times from $b(n)$, a time derivative of $b(n)$ is computed, $d(n)$, using a simple differencing operation (high-pass filtering) as given by $d(n) = b(n) - b(n-1)$. As shown in Fig. 2.2(c),

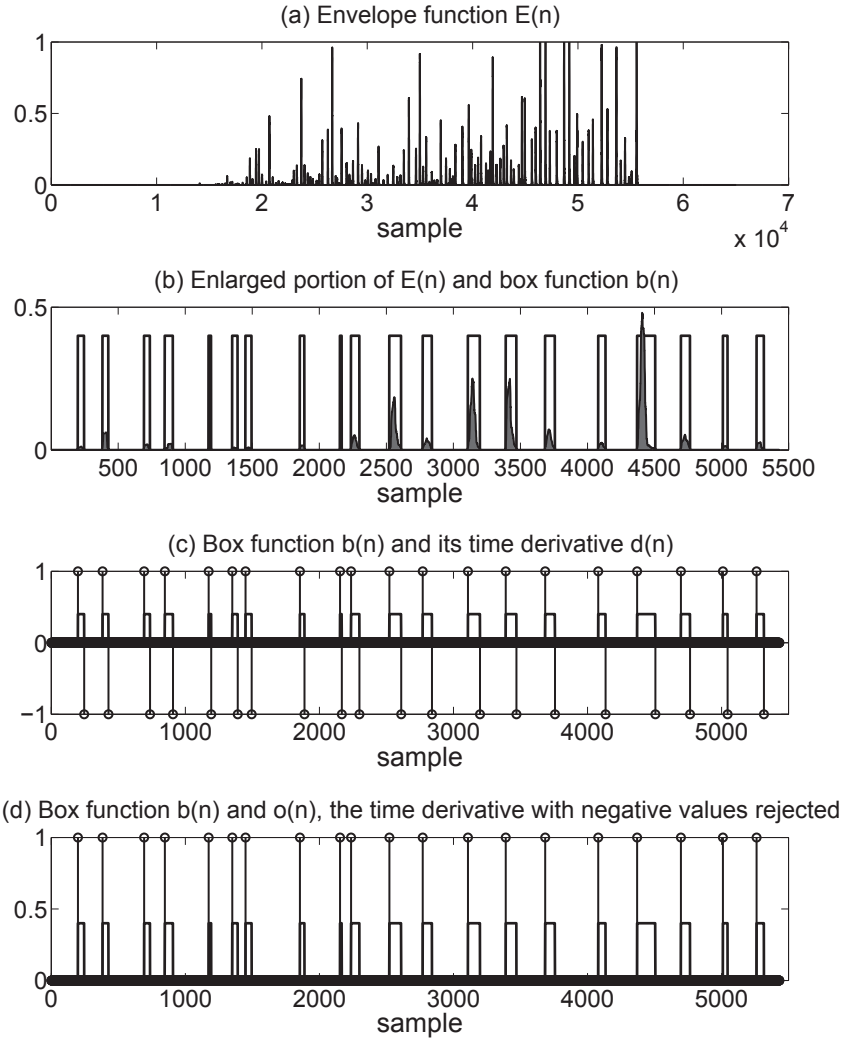

Fig. 2.2 (a) Envelope function $E(n)$ of given rolling sound $y(n)$. (b) Enlarged portion of $E(n)$ in (a) and its associated box function $b(n)$ (Eq. 2.2). (c) Box function $b(n)$ from (b) and $d(n)$, a time derivative of $b(n)$ (vertical lines) (Eq. 2.2), here $\alpha = 1$. (d) Box function $b(n)$ from (b) and $o(n)$ (vertical lines).

$d(n)$ contains values of either α or -α. Finally, by rejecting the negative values in $d(n)$, we obtain Eq. 2.3.

$$o(n) = \begin{cases} d(n) & \text{if} \quad d(n) > 0 \\ 0 & \text{otherwise.} \end{cases} \tag{2.3}$$

The function $o(n)$ is shown in Fig. 2.2(d), from which the contact timing index information function $i(k)$ is finally defined as

$$i(k) \quad : \quad \text{sample indices where } o(n) = \alpha \qquad (2.4)$$
$$k = 1, 2, 3, 4, \cdots, N_\alpha$$

where N_α is the total number of α in $o(n)$. From our basic assumption of the rolling dynamics, we wish to feed one contact sound at a time into the analysis/synthesis system. Thus, 'one contact sound' is defined as a segment of the original signal $y(n)$ whose length is the interval between two identified adjacent contact indices $i(k+1)-i(k)$. The kth contact sound $x_k(n)$ is defined as follows:

$$x_k(n) = y(n + i(k) - 1),\ n = 1, 2, ..., i(k+1) - i(k). \qquad (2.5)$$
$$k = 1, 2, \cdots, N_\alpha.$$

2.5 Analysis and Synthesis System

The analysis and synthesis scheme proposed here is devised to identify the excited modes of a single contact sound $x_k(n)$. An input segment $x_k(n)$ is first decomposed into subband signals by a tree structure filter bank. This not only improves the LP analysis by limiting the frequency range over which resonances are estimated, but it also allows for different LP parameters in each subband, which may be informed by perceptual characteristics. A notch detection operation for each segment is also performed to account for the time-varying, position-dependent suppression/attenuation of resonant modes as an object rolls over a surface.

2.5.1 Tree structure filter bank

A tree structure filter bank [25] is used to separate each contact segment into different frequency bands having unequal bandwidths for both the analysis and synthesis operations. The tree structure filter bank is constructed with two basic filters - one low-pass filter and another high-pass filter - and two-channel quadrature mirror filters (QMF) (Fig. 2.3, Fig. 2.4) are used to achieve a perfect reconstruction (PR) condition for the filter bank. Four filters of the QMF bank (two at the analysis bank (AB), $H_0(z)$, $H_1(z)$, and another two at the synthesis bank (SB), $G_0(z)$, $G_1(z)$) are related as below, whereby the alias-free property and the power symmetric condition are met [25]:

$$H_1(z) = H_0(-z),\ G_0(z) = H_0(z),\ G_1(z) = -H_1(z). \tag{2.6}$$

A 4-band structure was empirically chosen, with cut-off frequencies of $\frac{1}{8}\pi$,

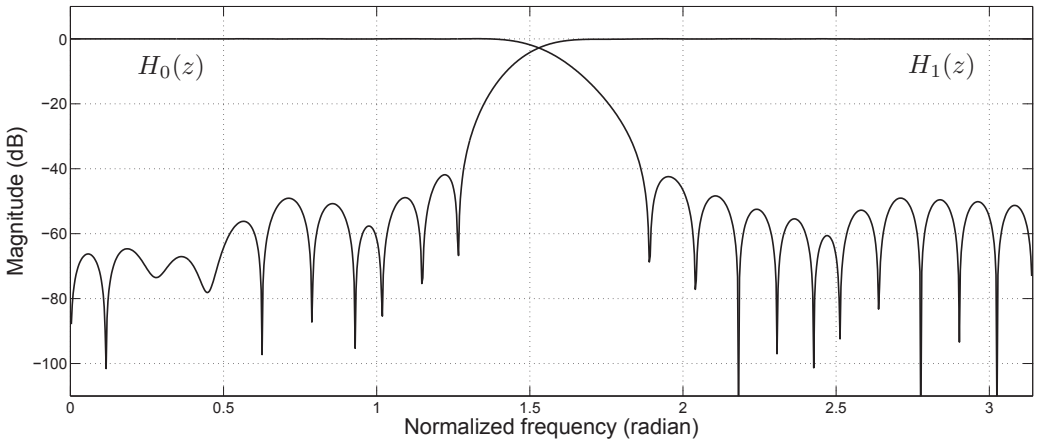

Fig. 2.3 Magnitude responses of a two channel QMF.

$\frac{1}{4}\pi$, $\frac{1}{2}\pi$ on a normalized frequency axis (Fig. 2.5). This filter bank can also be represented as a typical 4-band filter bank (Fig. 2.6) using the Noble identity [25]. Using the filters in Eq. 2.6, all the filters that constitute the analysis

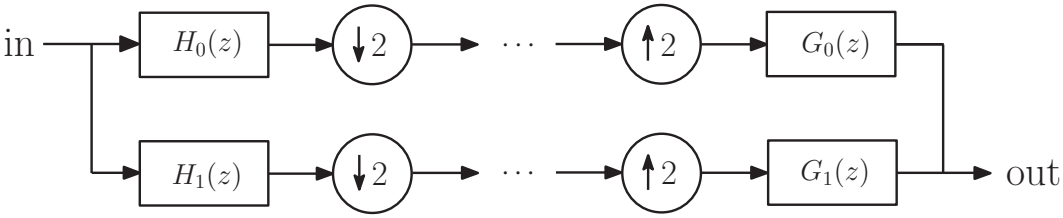

Fig. 2.4 Two channel QMF bank.

bank and the synthesis bank can be derived as given in Table 2.1

Analysis Bank	Synthesis Bank
$V_1(z) = H_0(z)H_0(z^2)H_0(z^4)$	$W_1(z) = G_0(z)G_0(z^2)G_0(z^4)$
$V_2(z) = H_0(z)H_0(z^2)H_1(z^4)$	$W_2(z) = G_0(z)G_0(z^2)G_1(z^4)$
$V_3(z) = H_0(z)H_1(z^2)$	$W_3(z) = G_0(z)G_1(z^2)$
$V_4(z) = H_1(z)$	$W_4(z) = G_1(z)$

Table 2.1 Transfer functions in the 4-band tree structure filter-bank

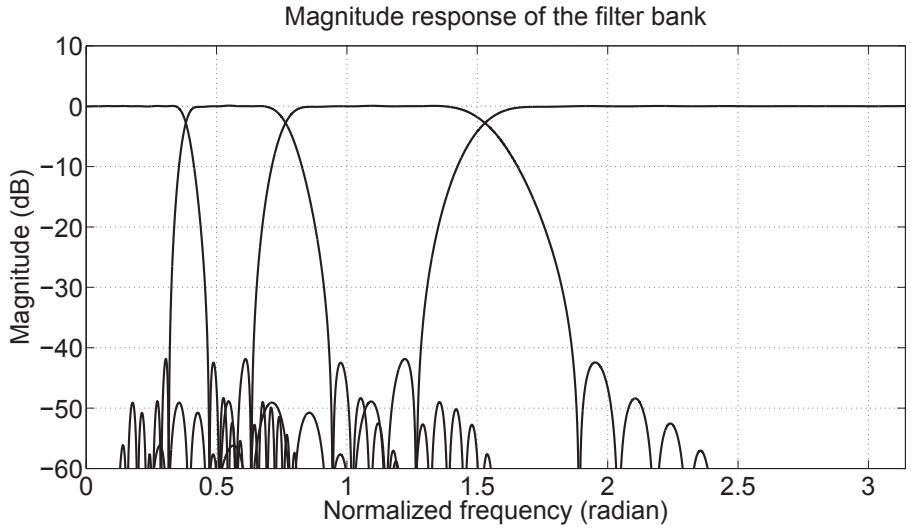

Fig. 2.5 Magnitude response of the filter bank.

In addition, the filter $H_0(z)$ is designed with a linear phase characteristic [26] so that the whole tree structure filter bank is piecewise linear phase. Therefore we are able to easily compensate for group delays introduced by the

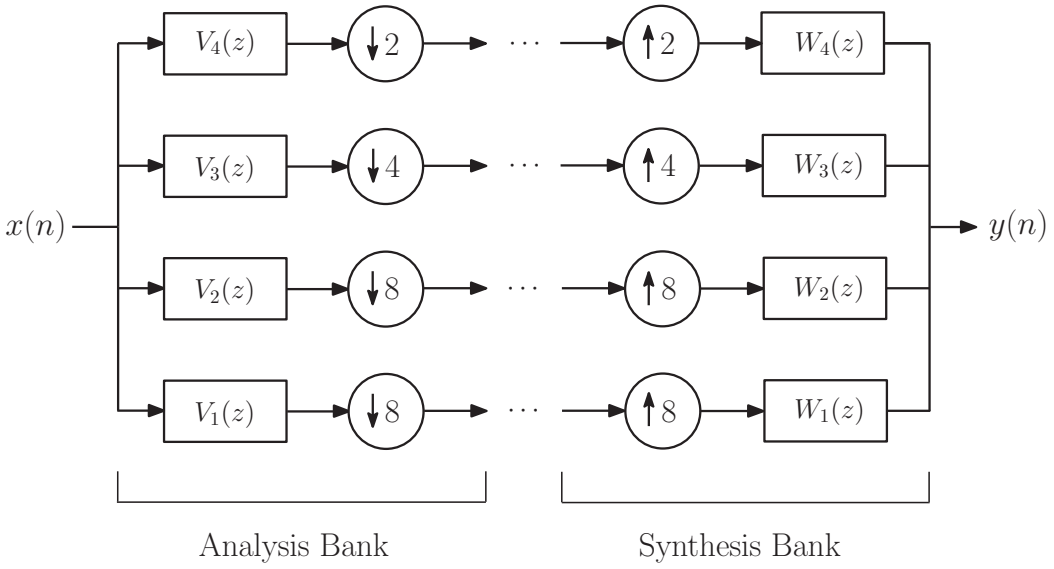

Fig. 2.6 Non-uniform 4-band filter.

filter bank through simple time-domain shifting.

2.5.2 Analysis/synthesis of one contact sound

When an object collides with a surface, the surface is set into motion by the excitation force exerted by the object. The vibrational motion of the surface is characterized by its modal properties, which in turn are determined by its geometry and physical characteristics. Because of the finite dimension of the surface, modes are selectively excited and attenuated or suppressed, depending on the location of the excitation. In a frequency magnitude response, excited modes appear as peaks, and suppressed modes appear as time-varying notch patterns that move in a self-consistent way over regions of the spectrum where energy was previously found. For example, Fig. 2.7 illustrates a simulated modal pattern for contacts along the length of a simply supported rectangular plate and a similar upwardly varying notch pattern in the spectrogram of $y(n)$. For each contact segment, we thus model both the time-varying spectral peaks (using LP) and the notches. In general, it is known that excited modes, which

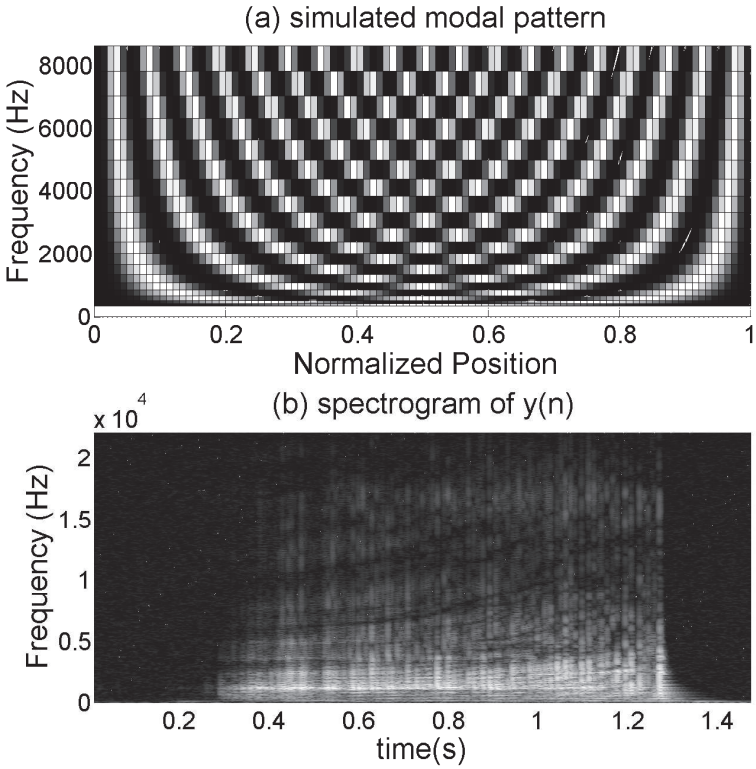

Fig. 2.7 (a) Simulated modal pattern of simply supported rectangular medium-density fiber (MDF) plate (Width: 0.95m, height: 0.25m, thickness: 0.02m) excited by a rolling object traveling from one end to the other end of the longer side, while centered on the other axis. (b) Spectrogram of $y(n)$. Upwardly varying notches can be seen.

represent resonances, are essential for the perception of sounds. However, in the case of rolling sounds on a finite-length rigid surface, existing notches in the spectrum are also important as their notch frequencies vary with time. In addition, by estimating notches, we can reduce the LP order (which would otherwise be unnecessarily high for describing zeros [27]).

Estimation of zeros using notch filtering

The kth input signal $x_k(n)$ is split into 4 subbands and downsampled at AB. Subband signals $x_k^{(l)}(n)$ are defined as follows:

$$x_k^{(l)}(n) = \textsc{Downsample}([v_l * x_k](n)) \tag{2.7}$$

where l is the order of the subband and $v_l(n)$ is the impulse response of the lth subband filter at AB of the filter bank. 'Downsample' and '$*$' denote downsampling and convolution operations, respectively.

Because the attenuated modes, as well as the excited modes, are perceptually important in characterizing the location of a rolling object, we are focusing on the estimation of the suppressed modes, represented as notches in the spectrum, as well as the excited modes.

In order to estimate the notches of $x_k^{(l)}(n)$, we considered building a notch filter where frequencies and bandwidths of notches would be modeled according to the valleys in the frequency response of $x_k(n)$. To this end, $|X_k^{(l)}(e^{j\omega})|$ (Fig. 2.8(a)), the magnitude of the Fourier Transform of $x_k^{(l)}(n)$, was flipped to $1/|X_k^{(l)}(e^{j\omega})|$ (Fig. 2.8(b)) and its peak frequencies,

$$\omega_{k,m}^{(l)} \qquad \text{for} \quad m = 1, 2, \ldots, M \tag{2.8}$$
$$\text{M : number of detected peaks}$$

were detected using the MATLAB function `findpeaks`. Then, by using quadratic polynomial curve fitting, lobes representing peaks were modeled to estimate 3dB-bandwidths $BW_{k,m}^{(l)}$ [2]. Once $\omega_{k,m}^{(l)}$ and $BW_{k,m}^{(l)}$ were estimated (in nor-

malized radian frequencies), we could form a zero as

$$z_{k,m}^{(l)} = e^{(-BW_{k,m}^{(l)}/2)} e^{-j\omega_{k,m}^{(l)}} \tag{2.9}$$

representing a valley in $|X_k^{(l)}(e^{j\omega})|$ [2]. Then biquad sections representing a suppressed mode were derived as

$$B_{k,m}^{(l)}(z) = (1 - z_{k,m}^{(l)} z^{-1})(1 - \overline{z}_{k,m}^{(l)} z^{-1}) \tag{2.10}$$

$$A_{k,m}^{(l)}(z) = (1 - \rho z_{k,m}^{(l)} z^{-1})(1 - \rho \overline{z}_{k,m}^{(l)} z^{-1}), \tag{2.11}$$

where $B_{k,m}^{(l)}(z)$ and $A_{k,m}^{(l)}(z)$ are the numerator and the denominator of the biquad section, respectively, $\rho = 0.95$, and $\overline{z}_{k,m}^{(l)}$ denotes the complex conjugate of $z_{k,m}^{(l)}$. $A_{k,m}^{(l)}$ plays the role of isolating each notch properly [28]. The notch filter $N_k^{(l)}(z)$ is given as follows:

$$N_k^{(l)}(z) = \prod_{m=1}^{M} \frac{B_{k,m}^{(l)}(z)}{A_{k,m}^{(l)}(z)}. \tag{2.12}$$

As shown in Fig. 2.8(c), the constructed notch filter has notches whose frequencies and bandwidths are the same as those of peaks in $1/|X_k^{(l)}(e^{j\omega})|$ (Fig. 2.8(b)). $X_k^{(l)}(z)$ was then filtered with $1/N_k^{(l)}(z)$ as below to obtain $Q_k^{(l)}(z)$:

$$Q_k^{(l)}(z) = \frac{X_k^{(l)}(z)}{N_k^{(l)}(z)}. \tag{2.13}$$

In $Q_k^{(l)}(e^{j\omega})$, notches are removed since $1/N_k^{(l)}(z)$ is an inverse filter of the notch filter, thus enabling LP estimation with lower orders (Fig. 2.8(d)).

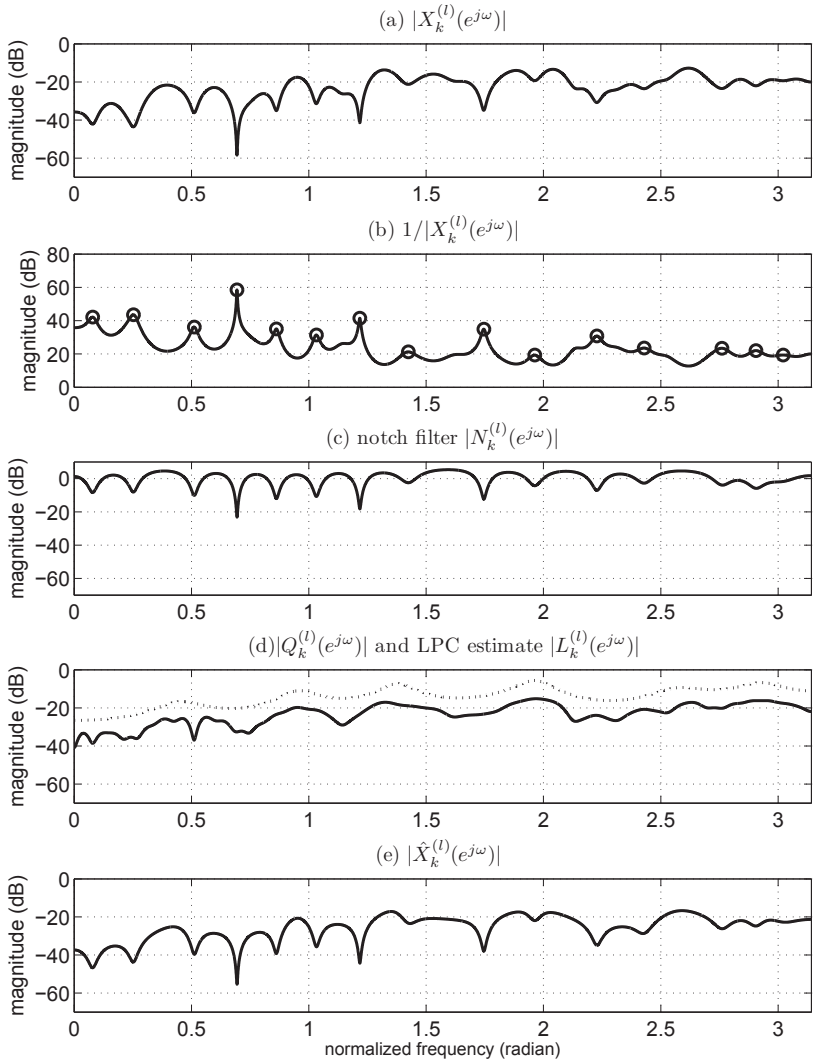

Fig. 2.8 (a) Magnitude of $X_k^{(l)}(e^{j\omega})$. (b) Magnitude of $1/X_k^{(l)}(e^{j\omega})$. Circle marks denote detected peaks. (c) Magnitude response of the notch filter $N_k^{(l)}(e^{j\omega})$. (d) Magnitudes of $Q_k^{(l)}(e^{j\omega})$ (solid) and its LP estimate $L_k^{(l)}(e^{j\omega})$ (dashed). (e) Magnitude of $\hat{X}_k^{(l)}(e^{j\omega})$. In all figures, x-axes denote normalized radian frequencies.

Estimation of poles using linear prediction

In order to estimate poles from $Q_k^{(l)}(z)$, a p_lth-order LP estimate $L_k^{(l)}(z)$ was derived as follows:

$$L_k^{(l)}(z) = \frac{G_k^{(l)}}{1 - \sum_{m=1}^{p_l} a_m^{(l,k)} z^{-m}}, \tag{2.14}$$

where $L_k^{(l)}(z)$ is the transfer function, $a_m^{(l,k)}$ are LP coefficients and $G_k^{(l)}$ is a gain of the LP estimate. $a_m^{(l,k)}$ were estimated in such a way that the LP error $e_k^{(l)}(n)$ as defined below was minimized [29]:

$$e_k^{(l)}(n) = q_k^{(l)}(n) - \sum_{m=1}^{p_l} a_m^{(l,k)} q_k^{(l)}(n-m), \tag{2.15}$$

where $q_k^{(l)}(n)$ is the impulse response of the $Q_k^{(l)}(z)$. $\hat{X}_k^{(l)}(z)$, the synthesis result of $X_k^l(z)$, was finally derived as (Fig. 2.8):

$$\hat{X}_k^{(l)}(z) = W_1(z)(\textsc{Upsample}(L_k^{(l)}(z)N_k^{(l)}(z))). \tag{2.16}$$

where Upsample denotes an upsampling operation. LP order p_l varies along subbands.

In Fig. 2.9, the LP estimates of the subband signals and the fullband signal used for the example in Fig. 2.8 are shown. All magnitude responses shown in Fig. 2.9 are without zero estimates applied. In the example of Fig. 2.8 and Fig. 2.9, the length of the contact sound $x_k(n)$ is 490 samples and the sampling rate is 44.1kHz. LP orders are set to $p_1 = 25$, $p_2 = 10$ and $p_3 = p_4 = 5$ for the subband signals and 45 for the fullband signal (no filter bank applied) so that the total orders of both cases are the same. It is clear that as a higher order is used for the low frequency region, significant spectral peaks are more effectively handled for a given total number of poles.

Since all the subband filters employed in the Synthesis bank are linear phase, their group delays $\tau_l^{(k)}$ are frequency independent, and only simple time-domain shifts arise as phase distortion. This well-behaved group delay property of the analysis/synthesis system is clearly evident in Fig. 2.10. Therefore, the phase distortion of $\hat{x}_k^{(l)}(n)$, the impulse response of $\hat{X}_k^{(l)}(z)$, can be

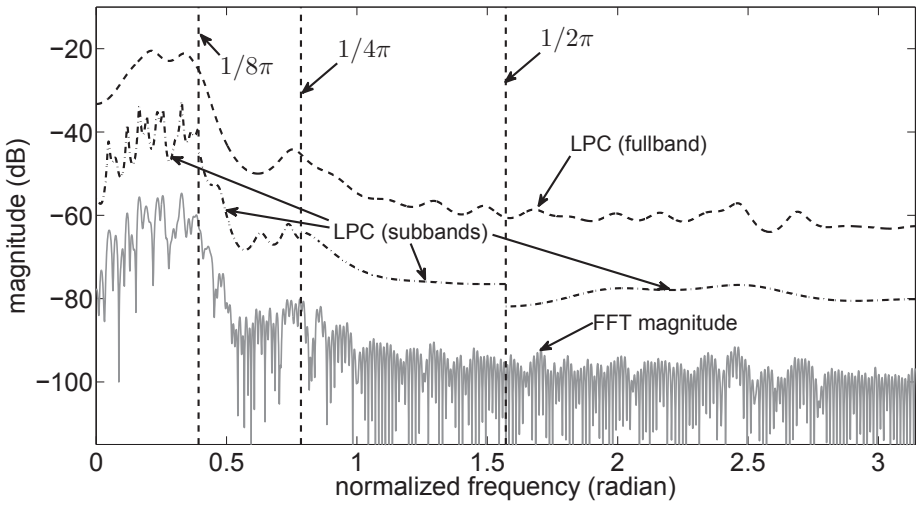

Fig. 2.9 Magnitudes of $X_k(z)$ and its syntheses. Gray line is the magnitude plot of $X_k(e^{j\omega})$ and black dotted line is the full band LPC estimate with order 45. Black dash-dotted lines are the magnitude responses of the LPC estimates of the subbands' signal from the lowest subband to the highest subband, respectively (zero estimates are not considered). LPC orders are 25, 10, 5, 5, from the lowest to the highest, respectively.

easily adjusted by shifting $\hat{x}_k^{(l)}(n)$ by $\tau_l^{(k)}$ which is estimated from the filter orders of $W_l(z)$. To complete the synthesis of $x_k(n)$, $\hat{x}_k^{(l)}(n)$ are shifted back

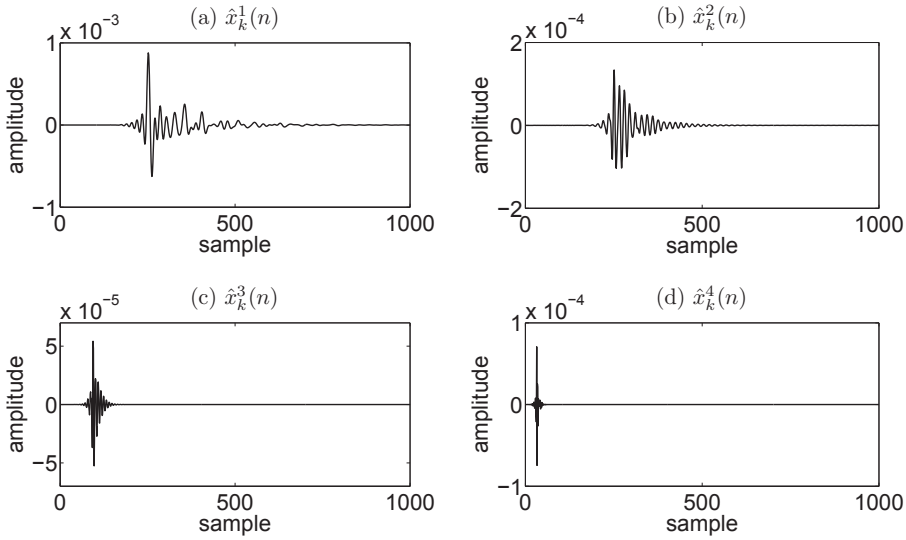

Fig. 2.10 Synthesized subband outputs ($\hat{x}_k^l(n)$): (a) $\hat{x}_k^{(1)}(n)$, (b) $\hat{x}_k^{(2)}(n)$, (c) $\hat{x}_k^{(3)}(n)$, (d) $\hat{x}_k^{(4)}(n)$.

by τ_l and added together as illustrated below to form s_k, the final synthesis output corresponding to $x_k(n)$.

$$\hat{x}_k^{(l)}(n) = \hat{x}_k^{(l)}(n + \tau_l^{(k)} - 1) \tag{2.17}$$

$$n = 0, 1, \cdots, n_k, \cdots, n_k + n_{extra}$$

$$s_k(n) = \sum_{l=1}^{num} \hat{x}_k^{(l)}(n) \tag{2.18}$$

num : total number of subbands.

where n_k is the length of $\hat{x}_k^{(l)}(n)$. As the transfer function of the synthesis result $\hat{X}_k^{(l)}(z)$ includes infinite impulse response (IIR) components, the impulse response $\hat{x}_k^{(l)}(n)$ must be truncated in order to make the length of $s_k(n)$ finite. $\hat{x}_k^{(l)}(n)$ is taken only up to n_{extra}, the number of samples determined empirically, and the samples proceeding from $n_{extra} + 1$ are discarded. The $s_k(n)$ segments are thus cascaded by using the overlap and add method in such a way that the location of $s_k(1)$ is matched to $i(k)$. Thus, the tail of $s_k(n)$ overlaps with a part of $s_{k+1}(n)$ and then they are added together.

2.6 Results

The proposed approach in this chapter demonstrates how to synthesize high-quality rolling sounds. However, the limits of this approach should be discussed. In general, rolling sounds vary greatly with respect to their associated physical aspects, such as surface roughness, ball speed, ball weight, size, etc. Our approach works well for rolling sounds where micro-contact events, which are presumed to correspond to microscopic collisions between the ball and the surface, are distinguishable when using onset detection-based techniques. On

the other hand, rolling sounds where the boundaries of micro contacts are ambiguous or rolling sounds that are too continuous for any micro contact event to be perceived would not be appropriate for the application of our approach. For such types of rolling sounds, segmentation based on micro contacts would result in randomly chopped sound fragments that would be physically and perceptually meaningless. Furthermore, the result of segmentation would dramatically vary even with a small adjustment of the configuration of the segmentation process. If these sound segments were used for synthesis, the resulting synthesized sounds would likely contain audible artifacts.

The rolling sound investigated in the previous sections is one example that can be re-synthesized well by using our method. As seen in the spectrogram in Fig. 2.1, micro contacts are well-separated and identifiable, facilitating the use of a high-pass filter that clearly reveals the boundaries of micro contacts. The rolling sound illustrated in Fig. 2.11 could be a counter example for which our approach is not as suitable as the example used thus far. This rolling sound is generated by rolling a steel ball on a steel plate. The high-pass filtered signal does not obviously reveal the impulsive components that represent the transients on which our analysis is based. This can also be observed in the spectrogram, which does not show 'vertical stripes' representing broadband transients in the high frequency region that are observable in the spectrogram of Fig. 2.1.

For the sake of comparison, we applied our method to this example. The high-pass filter used is the same type as the one used for the previous example but with a different cut-off frequency that was empirically set. Since no prominent transients were observed in the high-pass filtered signal, it was difficult to choose a proper N, the number of samples to be averaged, which is empir-

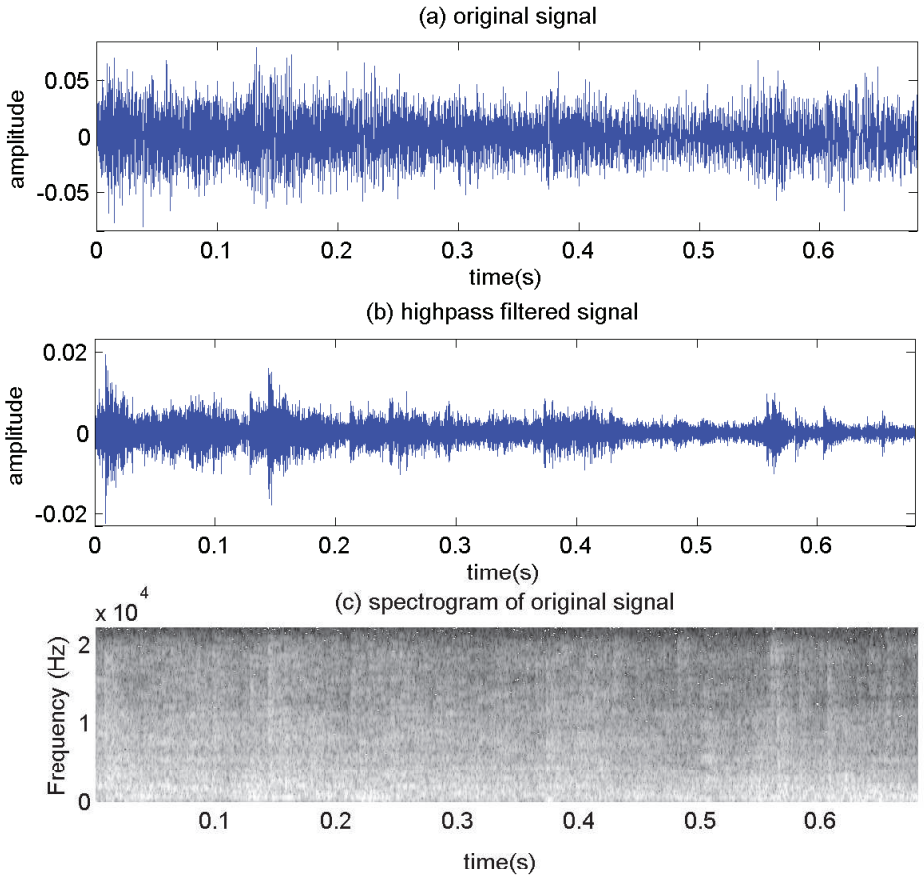

Fig. 2.11 Top : Original rolling sound. A steel ball rolling on a steel plate. Middle : High-pass filtered rolling sound. Bottom : Spectrogram of the original sound.

ically determined, for deriving the envelope function $E(n)$ and the threshold for deriving a box function $b(n)$. Thus $E(n)$ would be subject to yielding a box function $b(n)$ that would in turn result in segmentation in undesirable way. Fig. 2.12 shows an example in which there are too many peaks in $E(n)$, which results in many detected peaks $o(n)$ that do not seem to have correlations with micro-contacts, but rather look random.

On the other hand, Fig. 2.13 shows an analysis result where the peaks are sparse in $E(n)$ as more samples are averaged while deriving $E(n)$. Sparsely detected peaks, which are also not very likely to represent micro-contacts,

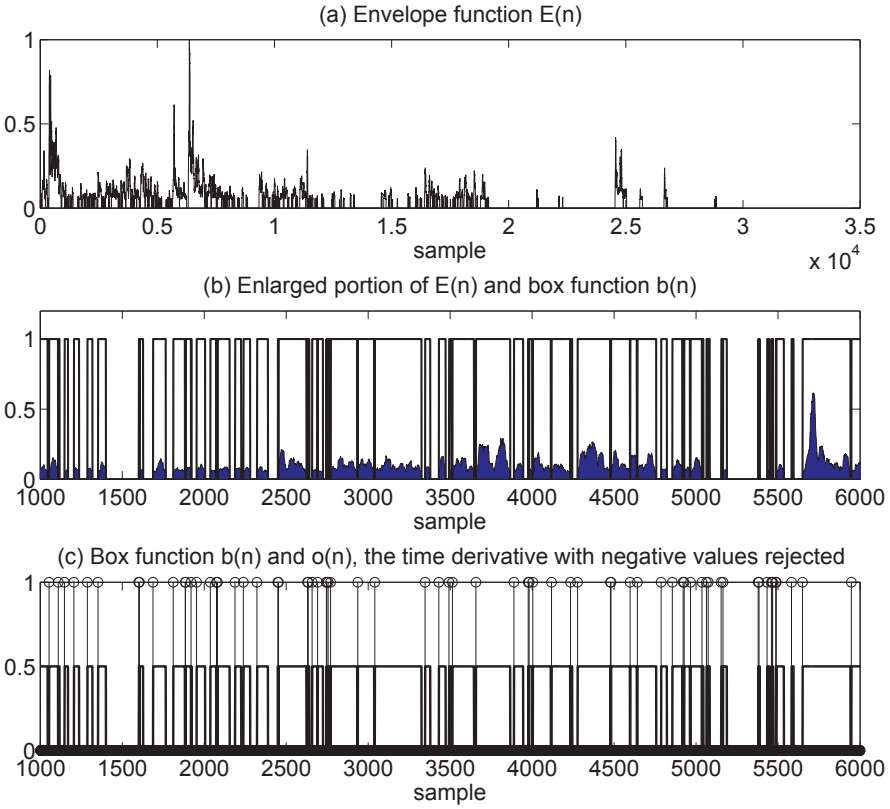

Fig. 2.12 (a) Envelope function $E(n)$ of a rolling sound generated by rolling a steel ball on a steel plate. (b) Enlarged portion of $E(n)$ in (a) and its associated box function $b(n)$. (c) Box function $b(n)$ from (b) and $o(n)$.

lead to a box function that is too sparse. The segmentation results from these two different analyses would not lead to successful re-synthesis of the original rolling sound as almost no correlation between micro-contacts and the segments is observed.[1]

[1]The re-synthesized sound examples based on the results of analyses on the counter sound example can be found on-line. `http://www.music.mcgill.ca/~lee/thesis/rolling`

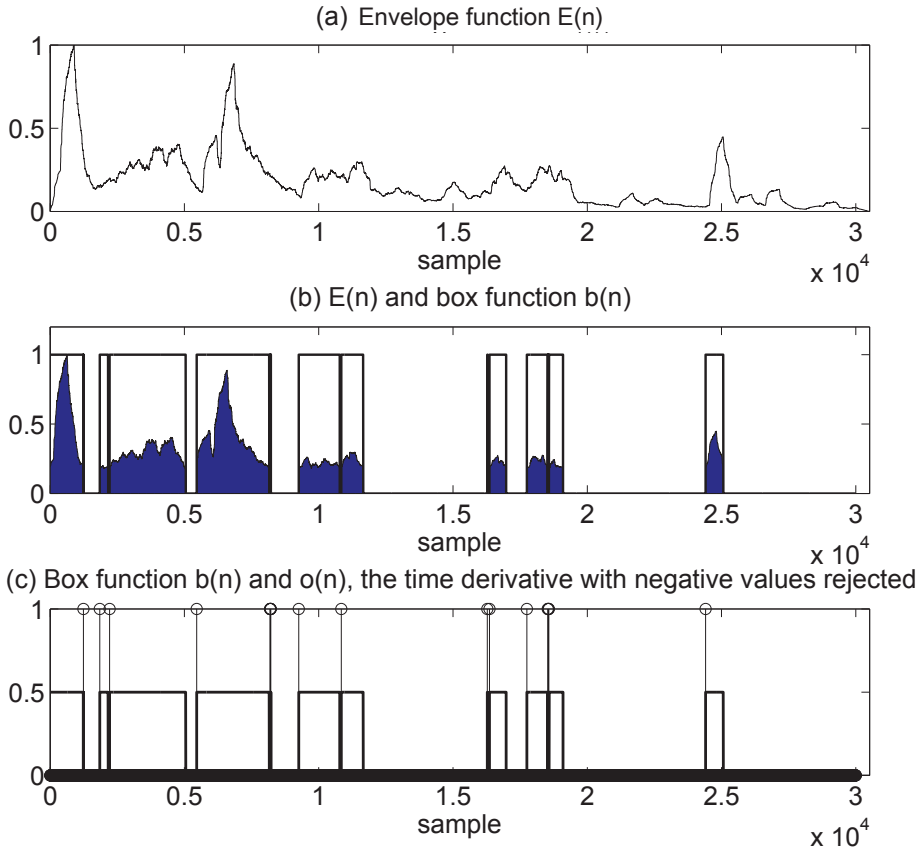

Fig. 2.13 (a) Envelope function $E(n)$ of a rolling sound generated by rolling a steel ball on a steel plate. (b) $E(n)$ in (a) and its associated box function $b(n)$. (c) Box function $b(n)$ from (b) and $o(n)$.

2.6.1 Synthesis Using Noise Signal Input

The counter example introduced above can be classified in a general group of sounds that lack strong impact-like events, such as scraping, rolling on a smooth surface, or gentle rain. As discussed above, the proposed approach is not likely to successfully resynthesize these kinds of sounds. Particularly, the sensation of smoothness, which is a common characteristic of these sounds, is hard to reproduce. Thus, as in an attempt to resynthesize this group of sounds, resynthesis using a noise signal as an input is investigated. First, a

sound is segmented in a way that the envelope function $E(n)$ has sparse peaks, as shown in Fig. 2.13, and then transfer functions representing resonances and anti-resonances in each subband estimated at the detected peaks are convolved with white noise signals to yield noise-driven resynthesis of the segments as follows:

$$s_{k,noise}(n) = \sum_{l=1}^{num}([\hat{x}_k^l * u](n)) \tag{2.19}$$

$$n = 0, 1, \cdots, n_k, \cdots, n_k + n_{extra}$$

$$num : \text{total number of subbands.}$$

where $u(n)$ is the noise signal and $s_{k,noise}(n)$ is the synthesized segment. The synthesized segments $s_{k,noise}(n)$ are then cascaded in the same way as in the previous approach. Since the length of the noise signals for each segment is the same as the intervals between segments, the synthesized sound does not give the sensation of undesired discontinuity. However, neither does it necessarily convey a sense of rolling, since it not only fails to distinguish itself from sounds such as those of scraping and "whooshing", but it also fails to update the transfer function often enough to reflect the location-dependent comb-filter effect, all of which is due to the sparseness of detected peaks in the analysis. As an attempt to overcome these artifacts, instead of using the detected peaks as the triggers to update the transfer function, we let the transfer function be updated periodically. The resynthesized sounds obtained by periodically updating the transfer function are found to maintain the comb-filter effect more effectively and sometimes better avoid confusion with the other kinds of smooth sounds, depending on the nature of the target sound and the periodicity set by the user, in comparison to the resynthesized sounds

based on the sparse peaks. However, the resulting sounds are quite sensitive to the periodicity set by the user. Sound examples of what we have discussed are available online at `http://www.music.mcgill.ca/~lee/thesis/rolling/`

Through investigating a counter example, we are able to get an idea of which types of rolling sounds our approach is suitable for, and an alternative approach is experimented with to devise an approach that would synthesize sounds of the same type as the counter example.

2.7 Conclusion

An analysis and synthesis approach for rolling sounds is proposed in this chapter. The process is based on the assumption that an overall sound can be linearly decomposed into many micro-contacts between an object and the surface on which it rolls. Therefore, a process similar to onset detection is carried out to extract the contact timing information and to segment the sound into individual contact events. Each segment is fed into an analysis/synthesis system to estimate time-varying filters. The analysis/synthesis process consists of a tree structure filter bank, LP processors and notch filters. Thanks to the tree structure filter bank, LP orders can be flexibly assigned to subbands, allowing us to focus more on significant spectral features while analyzing and synthesizing with LP processors and notch filters. The resynthesized contact events are appropriately cascaded by using the overlap and add method to produce the final rolling sound. We find that the performance of the proposed approach depends on the characteristics of rolling sounds. The proposed approach works better for rolling sounds in which each micro contact is relatively well separated and so identifiable, rather than for those that are sonically smooth and continuous. Also, an alternative method that can effectively synthesize the

sounds for which the proposed approach is not suitable is discussed.

Chapter 3

Granular Analysis/Synthesis for Simple and Robust Transformations of Complex Sounds

3.1 Introduction

In this chapter, a novel granular analysis/synthesis system particularly geared towards environmental sounds is presented. A granular analysis component and a grain synthesis component were intended to be implemented separately so as to achieve more flexibility. The grain analysis component would segment a given sound into many 'grains' that are believed to be microscopic units that define an overall sound. A grain is likely to account for a local sound event generated from a microscopic interaction between objects, for example, a sound of a single water drop hitting the ground in the sound of rain. Segmentation should be able to successfully isolate these local sound events in a physically or

perceptually meaningful way, with only a few easy-to-understand parameters for user convenience. The second part of the research was focused on the granular synthesis that can easily modify and re-create a given sound. The granular synthesis system would feature flexible time modification with which the user could re-assign the timing of grains and adjust the time-scale. Also, the system would be capable of cross-synthesis given the target sound and the collection of grains obtained through an analysis of sounds that might not include grains from the target one.

3.2 Background

Nowadays, audio rendering in virtual reality applications, especially games, requires higher standards to meet the users' demands. Conventional ways of sound generation in games, mostly playing pre-recorded samples, are often limited in their lack of ability to deal with variations in sounds, for interactions between objects in games occur in various ways. This problem demands model-based sound synthesis techniques capable of generating many sound instance variations without having to use additional sound samples. Sounds that appear in games are in general non-musical/verbal, often referred to as 'environmental' or 'everyday' sounds. Such sounds are generated mostly either from interactions between objects or environmental background that is given in the virtual space, including bouncing, breaking, scratching, rolling, streaming, etc. It is very important to maintain the quality of such sounds for a feeling of reality. In general, every synthesis technique has its own strength and it differs according to the types of sounds. Therefore it is crucial to choose a synthesis technique appropriately, given the sounds to be dealt with. The granular analysis/synthesis technique is regarded as one of the promising methods to

deal with sounds in games since the technique can easily preserve complex sound textures and create variations of the given sound by mosaicking grains. Thus, it would be helpful to develop a novel granular analysis-based synthesis framework that could be used easily by non-signal processing experts to allow parametric synthesis controls to generate many variations of complex sounds. This framework could use the benefits of granular synthesis to fill the gap between information contained locally in the waveform (specific to a grain) and global information about the sound production process, such as resonance frequencies. It could also use information derived from an understanding of physical processes to control the density and time-distribution of sound grains.

The concept of granular analysis/synthesis dates back to Dennis Gabor who proposed the idea of granular synthesis to represent sound by using sound snippets generally shorter than a typical musical note [30]. The Greek composer Iannis Xenakis was attracted to this theory and investigated compositional theory based on sound grains [31]. Curtis Roads is regarded as the first person to implement granular synthesis using digital computers [32]. Barry Truax first implemented real time granular synthesis using digital signal processors [33].

The idea of granular analysis/synthesis has been extended to explore analysis/synthesis of general complex sounds, such as environmental sounds. Inspired by the unit-selection based text-to-speech synthesis research in [34], Schwarz and his collaborators have conducted research on concatenative-based synthesis to generate not only musical but also abstract complex sounds, aiming at synthesizing sound textures in particular [35] [10], referred to as 'Corpus-based concatenative synthesis'. A database of short sound snippets, the 'corpus', is first constructed by segmenting recorded audio sounds. The database

also contains data of particular features associated with each sound snippet. Features extracted range from the low-level properties of an audio phenomenon to the high-level descriptors. Additional variants of existing sound snippets created by transformations of original sound snippets, are also available in the database. Thanks to a wide variety of features, a sound can be synthesized in various feature-matching ways.

Lu *et al.* [36] attempted to generate 'audio texture' by concatenating pre-segmented short sound snippets from recorded sounds. The transition probability between sound snippets was first derived from the Mel Frequency Cepstral Coefficients (MFCC) of snippets and the sequence of snippets was determined according to the similarity score based on the transition probability of certain conditions that prevent the synthesis from being audibly uncomfortable. Picard *et al.* [11] used a dictionary of short sound segments analyzed from recorded sounds, and synthesized a sound with respect to a given target sound by selecting best matched sound segments in a dictionary with time modification. They also proposed a way to concatenatively synthesize a sound with the segment selection informed by a physics engine. A work by Dobashi *et al.* [37] aimed at synthesizing aerodynamic sounds in a concatenative manner. Sound segments constituting a database were not extracted from recorded sounds but were created using physical model-based simulation. Hoskinson and Pai [38] used an algorithm that involves wavelets to segment an input audio signal into 'syllable-like pieces', and synthesis was conducted by selecting segments according to a similarity measure and concatenating them.

Granular analysis/synthesis environments, with GUIs, have also been developed by several researchers, as found in [39], [40], [41] [42]. They serve as tools that enable users to intuitively and interactively synthesize sounds for

various applications using the granular analysis/synthesis framework.

The goal of this research is to flexibly synthesize complex sounds within the framework of granular analysis/synthesis. To this end, not only existing granular analysis/synthesis techniques are enhanced and customized, but also novel features are introduced, culminating in a novel granular analysis/synthesis scheme. In the analysis stage, a measure referred to as the 'stationarity' is proposed to categorize a complex sound into the region where distinctive micro sound events can be identified (non-stationary region) and the region where the boundaries of the micro sound events are too ambiguous to be distinguished from each other (stationary region). This is done to adjust parameters associated with granular analysis so as to achieve more promising granular synthesis. In the synthesis stage, to enable flexible synthesis of complex sounds aiming at various kinds of interaction scenarios, time modification allows for seamless time stretching and shrinking with the aid of proposed gap-filling algorithms and also for grain time remapping by re-ordering the locations of grains, which leads to modifying given complex sounds in a physically meaningful way.

3.3 Granular Analysis System

The granular analysis system involves the decomposition of a sound into short snippets, termed as the 'grains', on the assumption that a sound is generated from numerous micro-interactions between physical objects. For example, the sound of rain consists of sounds of collisions of innumerable rain drops on other objects. The grain analysis aims at segmenting a sound into grains, using a proper segmentation technique, in a way that the segmented grains are associated as much as possible with microscopic events such as physical collisions. In the granular analysis system proposed here, analysis begins by

carrying out a process similar to onset/transient detection for segmenting a sound composed of grains that have percussive/impulsive characteristics. The grain analysis system is implemented in MATLAB with a GUI where one can set all the parameters (Fig. 3.1). The parameters used for analysis are listed in Table 3.1.

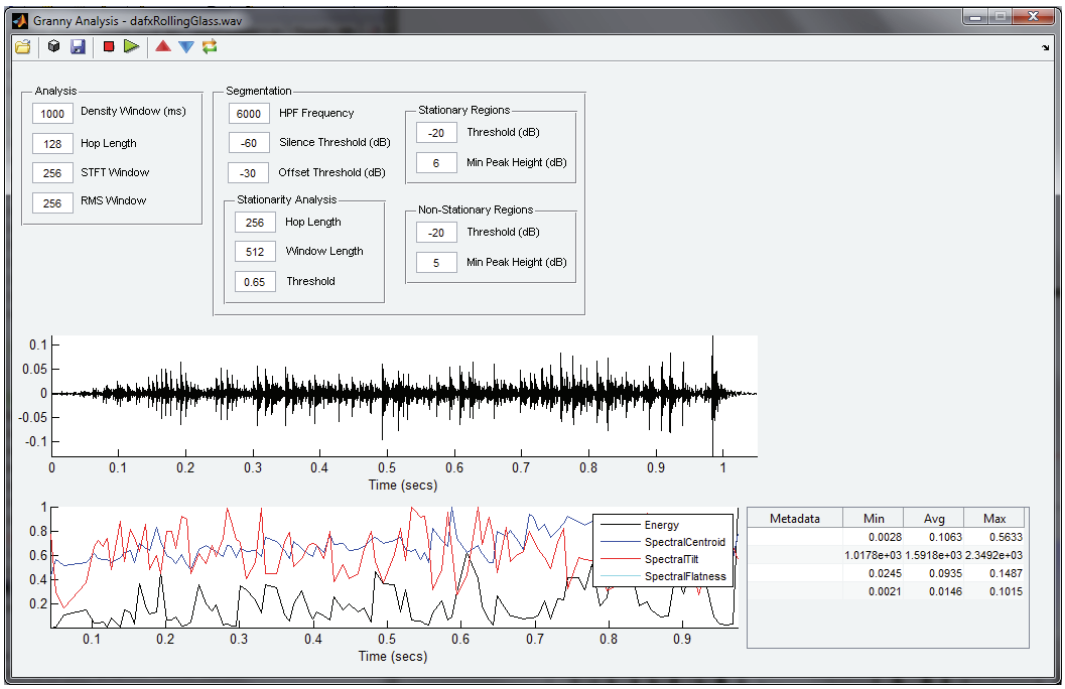

Fig. 3.1 GUI for granular analysis.

3.3.1 Grain Analysis

In order to perform the task of granular analysis, it is essential to transform an audio signal into a form that reveals and emphasizes the transients in the audio signal, referred to as the *detection function* [24]. Many detection functions have been devised and used to effectively analyze audio signals, using either time-domain or frequency-domain techniques. No dominant detection function that outperforms other detection functions exists, so a detection function is chosen and used depending on the nature of the given audio signal and the purpose

Granular analysis parameters		
Spectral Flux analysis	Density window Hop length STFT window RMS window	
Segmentation	HPF frequency Silence threshold Offset threshold	
	Stationary Analysis	Hop length Window length Threshold
	Stationary regions	Threshold Min peak height
	Non-stationary regions	Threshold Min peak height

Table 3.1 Granular analysis parameters.

of the analysis. In this research, we have chosen to use a detection function that measures differences in the spectral content of transient and non-transient parts of the signal.

The well-known short-time Fourier transform (STFT) [43] for frame-by-frame analysis enables comparison of spectral content between sequential, neighboring short portions of the signal. The STFT of $x(n)$ is given as

$$X_k(n) = \sum_{m=0}^{N-1} x(hn+m)w(m)e^{-2j\pi mk/N}, \qquad n = 0, 1, 2, \cdots \tag{3.1}$$

where h ('Hop Length' in Fig. 3.1 and Table. 3.1) is the hop length and w is a window of length N('STFT Window' in Fig. 3.1 and Table 3.1). $X_k(n)$ is the kth discrete Fourier transform (DFT) coefficient of the nth frame. There are many ways of detecting transients based on comparing the spectral content of successive frames. The Spectral Flux (SF) was chosen in this research because internal experiments showed it to be suitable for present purposes. The SF is

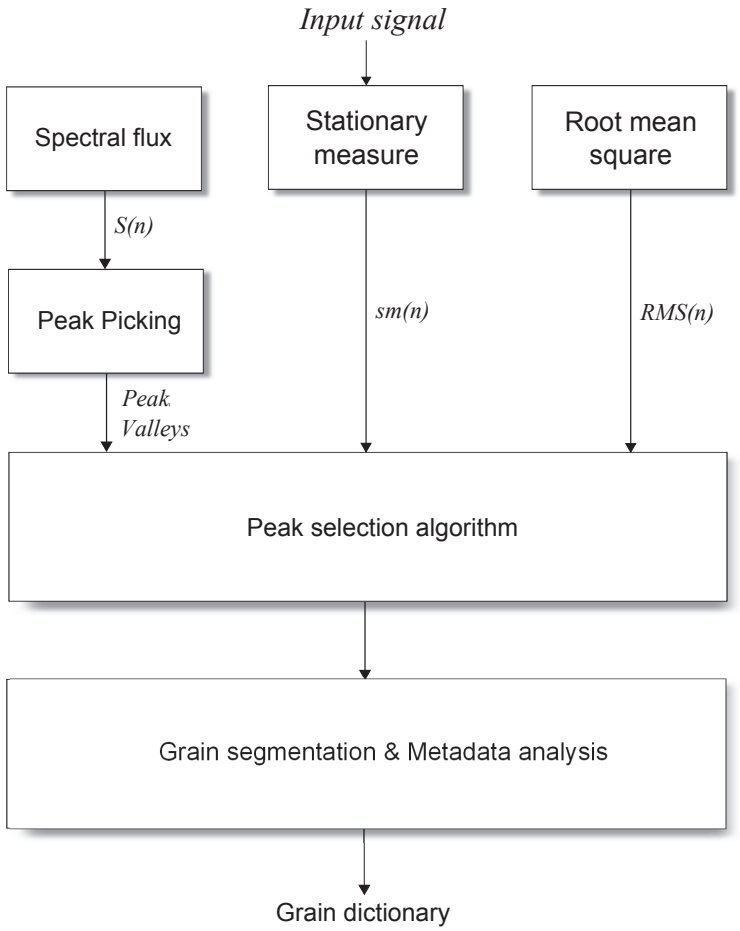

Fig. 3.2 Overview of granular analysis system. $S(n)$, $sm(n)$ and $RMS(n)$ are defined in Eqs. 3.2, 3.7 and 3.5, respectively.

defined as below [24],

$$S(n) = \sum_{k=0}^{N-1} \{H(|X_k(n)| - |X_k(n-1)|)\}^2 \tag{3.2}$$

where $S(n)$ is the value of the SF at the nth frame. $H(x) = \frac{(x+|x|)}{2}$ is a half-wave rectifier employed to put an emphasis on only the positive changes. The SF first measures the Euclidian distance between magnitude spectra of successive frames and takes into account only the energy increases in frequencies. In this way, the comparison between the magnitude spectra that have

a broadband energy and those that have a narrowband energy returns a high score, effectively detecting transient events in $x(n)$, as can be seen in Fig. 3.3. Depending on the nature of the signal to be analyzed, high-pass filtering can be conducted to reveal transients more vividly [17]. The parameter referred to as the 'HPF frequency' (Fig. 3.1 and Table 3.1) actually defines the cut-off frequency of the high-pass filter.

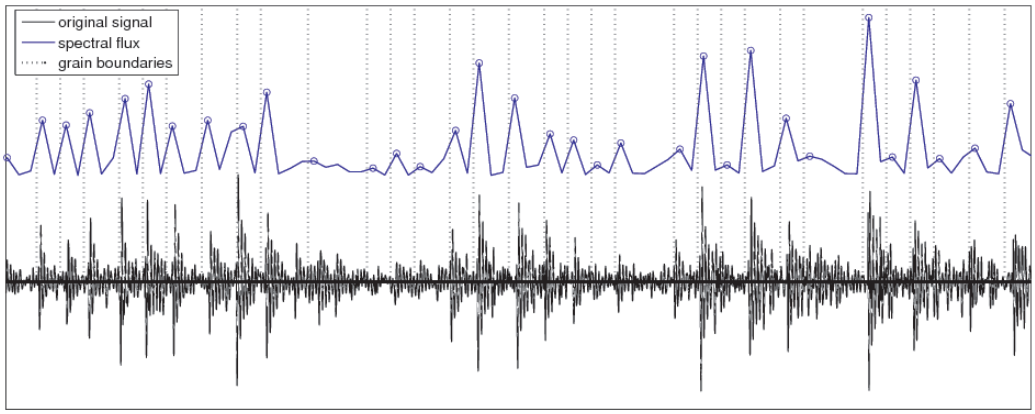

Fig. 3.3 Signal and spectral flux (SF).

3.3.2 Grain Segmentation

Peak Detection

Since a typical impact sound begins with a transient, broadband energy and then continues with decaying resonances, peaks in the SF are likely to appear around the beginning point of a local impulsive event. A peak in the SF, $S(n_{peak})$, is defined as

$$S(n_{peak} - 1) \leq S(n_{peak}) \geq S(n_{peak} + 1) \tag{3.3}$$

where n_{peak} is a sample index on which the peak is located. A valley in the SF, $S(n_{valley})$, is defined as,

$$S(n_{valley} - 1) \geq S(n_{valley}) \leq S(n_{valley} + 1) \tag{3.4}$$

where n_{valley} is a sample index on which the valley is located. Prior to conducting peak selection, noise components in the SF that may be confused as meaningful peaks are first discarded by partitioning the overall signal into silent/non-silent regions. In order to do this, we propose a parameter referred to as the 'silent threshold' (given in the 'Segmentation' category as in Fig. 3.1, Table 3.1) and calculate the frame-based short time root mean square (RMS) of the overall signal as

$$RMS(n) = \sqrt{\frac{1}{N_{rms}} \sum_{m=0}^{N_{rms}-1} |x(h_{rms} \cdot n + m)|^2} \tag{3.5}$$

where N_{rms}, h_{rms} is the length of the frame and the hop length used, respectively (the 'RMS window' the 'Hop Length' in the 'Analysis' category as in Fig. 3.1 and Table 3.1. As the $RMS(n)$ can be regarded as the absolute value of the roughly estimated amplitude envelope, regions where $RMS(n)$ are smaller than the silent threshold could be labeled silent regions. In this way, we could consider peaks only in the non-silent regions and reduce the chances of including unnecessary noise components. Also, a parameter called the 'Peak Threshold Height' (given as the 'Threshold' in the categories of 'Non-Stationary Regions' and 'Stationary Regions' (Fig. 3.1 and Table 3.1)) is defined to rule out peaks whose heights are lower than this parameter, making it possible to ignore peaks whose heights appear to be too small, depending on the nature of the given signal. By using the peak detection method proposed

in [44], peaks that satisfy a certain condition are picked to determine the grain segmentation. That condition is associated with a ratio γ in such a way that:

$$\gamma < \frac{S(n_{peak})}{(S(n^l_{valley}) + S(n^r_{valley}))/2} \tag{3.6}$$

where $S(n_{peak})$ is the SF value at the peak location and $S(n^l_{valley})$ and $S(n^r_{valley})$ are the SF value at the neighboring valleys to the left and right sides of the peak $S(n_{peak})$. The ratio γ is referred to as the 'Minimum Peak Height' (given as 'Min Peak Height' in the categories of 'Non-Stationary Regions' and the 'Stationary Regions' (Fig. 3.1 and Table 3.1)). The right hand side simply represents the ratio of the peak SF value and the average SF value of the two neighboring valleys. Only when the ratio of the peak and the valleys is larger than the minimum peak height, is $S(n_{peak})$ chosen as the grain segmentation boundary. A grain segmentation boundary is set in such a way that the beginning point of a grain is set at $n_{peak} - \frac{h}{2}$, a sample index ahead of the peak location by half of the hop length, to consider rising time in the attack phase of an impulsive event, as shown in Fig. 3.3.

Stationarity Analysis

Environmental sounds are often stationary in the sense that no distinctive events occur, but relatively consistent 'texture' is present. For example, as opposed to the sound of glass breaking which could be regarded as a 'non-stationary' sound inasmuch as relatively distinctive sound events constitute the overall sound, the sound of a gentle brook would convey more consistent and regular impressions to listeners. The stationary sounds usually consist of numerous sound events of very short durations heavily blended with each other so that an individual sound event is scarcely identifiable in the overall

sound. In this case, attempts to decompose a given sound into micro-sound events would not be efficient since the resulting segmented grains would tend not to correspond to meaningful sound events. For synthesis, grain segmentation can be performed on the relaxed condition that leads to grains with longer duration and preserves a sound texture rather than micro-events. The resulting synthesis would not be very different from the one based on normal grain segmentation in terms of how audibly natural it would seem. Thus it is desirable to be able to adjust criteria for grain segmentation according to the 'stationarity' of a given sound. To achieve this, we first propose a measure to detect which part of the signal is stationary or non-stationary. Here we assume that stationarity is closely related to how a signal looks in the time domain, in such a way that a stationary part would look statistically flat while a non-stationary signal would look rather 'bumpy'. The measure proposed is referred to as the 'stationarity measure' and is defined as,

$$sm(n) = \frac{\sqrt[N_{sm}]{\prod_{m=0}^{N_{sm}-1} |x(h_{sm} \cdot n + m)|}}{\frac{1}{N_{sm}} \sum_{m=0}^{N_{sm}-1} |x(h_{sm} \cdot n + m)|} \tag{3.7}$$

where h_{sm} is the hop length and N_{sm} (given as the 'Hop Length' and the 'Window Length' in the 'Stationary analysis' category in Fig. 3.1 and Table 3.1) is the frame size. The numerator and the denominator are the geometric mean and the arithmetic mean, respectively, of the absolute values of the samples contained in the nth frame. This can be viewed as the time domain version of the 'spectral flatness measure' [45]. The spectral flatness measure was originally devised to define the characteristics of an audio spectrum as either tonal or non-tonal by measuring the number of spiky components present in a power spectrum. The more peaks the spectrum contains, the more tonal

it will be, while a statistically flat spectrum indicates that the signal is non-tonal, or noise-like. The spectral flatness measure corresponds to the extent that a signal is bumpy in the time domain, as indicated by $sm(n)$. Once stationary/non-stationary parts are partitioned, we can individually set the parameters associated with peak detection, so that there are tighter conditions for non-stationary parts and more relaxed ones for stationary parts, as separated into the 'Non-stationary' and the 'Stationary' categories in Fig. 3.1 and Table 3.1.

The stationarity measure is sensitive to the window length as it determines the density of micro events. If the window length is large enough for the density of events to look as if they were 'flat' in the time domain, then the signal will be observed as if it were more stationary. If the stationary measure $sm(n)$ is above the stationarity threshold, then it will simply be labeled stationary.

Figure 3.4 shows an example of partitioning a signal into stationary/non-stationary regions on the basis of the stationarity measure. The signal in the top pane consists of two types of applause sounds. The one on the left of the blue dashed vertical line in the middle is applause by a large number of people, while the one to the right of the blue dashed vertical line is by a small number of people. The middle pane shows the $sm(n)$, and it is obvious that the applause by the large audience has a higher and consistent $sm(n)$, whereas the one by the small audience has a lower and varying $sm(n)$. The bottom pane shows that, on the basis of the stationarity measure, the parameters for peak detection can be separately set and yield different grain segmentation results accordingly.

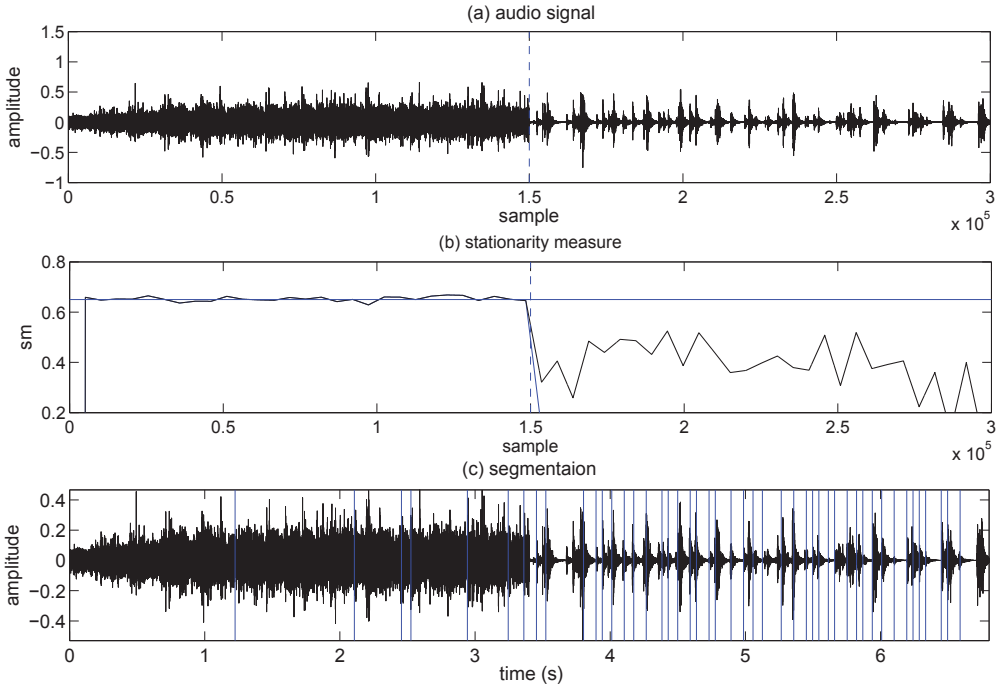

Fig. 3.4 Comparison of stationarity measure depending on the nature of a signal. (a) Original signal. The signal consists of two types of applause sounds. The one on the left of the blue dashed vertical line in the middle is applause by a large audience, while the one to the right of the blue dashed vertical line is by a small audience. (b) Stationarity measure of the signal in (a). The blue horizontal line is the silent threshold, set as 0.65. The hop length and the window length used are 6144 and 1024, respectively. (c) The result of grain segmentation with respect to two different sets of parameters. For the stationary part, the left side, the peak height threshold is -45dB and the minimum peak height is 11dB, and those for the non-stationary part, the right side, are respectively -25dB, 3dB.

3.3.3 Meta Data

For further applications, such as feature matching-based synthesis [40], meta data associated with a grain $g_k(n)$ (kth grain in the dictionary), features widely used in music information retrieval (MIR) and psychoacoustics research, are extracted as auxiliary information. Selected features that constitute meta data are the following (they are all normalized between 0 to 1).

- *Energy*

$$en(k) = \sum_{m=0}^{l_k} |g_k(m)|^2 \tag{3.8}$$

l_k : kth grain's length

- *Spectral Centroid*

$$sc(k) = \frac{\sum_{m=0}^{N-1} m|G_k(m)|}{\sum_{m=0}^{N-1} |G_k(m)|} \tag{3.9}$$

$G_k(m)$: mth DFT coefficient of g_k

N : DFT length

- *Spectral Tilt*

$$st(k) = \frac{N\sum_{m=0}^{N-1} m|G_k(m)| - \sum_{m=0}^{N-1} m \cdot \sum_{m'=0}^{N-1} |G_k(m')|}{N\sum_{m=0}^{N-1} m^2 - \left(\sum_{m=0}^{N-1} m\right)^2} \tag{3.10}$$

- *Spectral Flatness*

$$sfl(k) = \frac{\sqrt[N]{\prod_{m=0}^{N-1} |G_k(m)|}}{\frac{1}{N}\sum_{m=0}^{N-1} |G_k(m)|} \tag{3.11}$$

3.3.4 Grain Dictionary

All segmented grains are separately stored in a grain dictionary together with the meta data. In many cases, a grain has a long tail with small amplitude. We can set the parameter referred to as the 'Offset threshold' (Fig. 3.1 and Table 3.1), which defines the amplitude threshold below which the tail is discarded, to efficiently compress the size of the grain wave data. In the grain dictionary, an element that represents a grain contains the following:

- grain wave data
- the starting point and the end point in the original signal
- meta data
- sampling rate.

As will be explained later, the starting point and the end point data enable time modification at the synthesis stage.

3.4 Granular Synthesis

With the granular synthesis system we have developed, the user can flexibly manipulate the temporal aspects of a sound dictionary. With a sound dictionary given, a user can perform time-scaling (stretching/shrinking) and can shuffle grains at will. To this end, algorithms that can fill gaps which inevitably arise when grain timings are modified have been devised. One such algorithm is based on a signal extrapolation technique. This algorithm extends the grain just before a gap by using linear prediction (LP), extrapolating the grain in the manner of the source-filter synthesis approach, to fill the gap. Another gap filling scheme is achieved by choosing and inserting additional grains in the gap. To select additional grains that would result in perceptually natural synthesis, Mel-frequency cepstrum coefficients (MFCC) and Itakura-Saito (IS) distance were used to find additional grains that are perceptually similar to the grain just before a gap. A grain that has the minimum Euclidean distance of MFCC or the minimum IS distance with respect to the given grain just before a the gap is selected and put in the gap. A range from which an additional grain is selected is usually set around the given grain. The grain search continues until the gap is completely filled. As the algorithm based on

inserting additional grains is subject to selecting the same grain repeatedly for a single gap, which would result in an audible artifact, the system is designed to allow the user to randomly select an additional grain among the best candidates, instead of selecting the very best grain. The two different ways of gap filling, extrapolation and employing additional grains, result in different effects in synthesis. The extrapolation-based algorithm changes the grain density as time is re-scaled, whereas the additional grain-based algorithm tends to preserve the grain density. This property contributes to the flexibility of the synthesis system. In all the gap filling algorithms, grains can be tapered with symmetric/asymmetric windows and overlapped to avoid jitter-like artifacts.

As shown in Fig. 3.5, the grain synthesis system consists of three components. In the grain dictionary component, the user can load target and corpus grain dictionaries from which grains are selected for the synthesis. Once the grain dictionary is loaded, the user can manipulate the temporal length of the given sound by stretching or shrinking intervals between grains by adjusting parameters that belong to the time stretching component. The component of time scrub allows for more flexible time modification in conjunction with time stretching, by enabling users to re-arrange the original order of the grain sequence in the dictionary.

3.4.1 Grain Dictionaries: Target and Corpus

The granular synthesis system requires two types of dictionaries for synthesis. One is the target dictionary and the other is the corpus dictionary. In our granular analysis/synthesis system, the target dictionary provides the time position information of grains as a target reference. Let $i(k)$ denote the starting time sample index of the kth grain, g_k, in the target dictionary. The initial op-

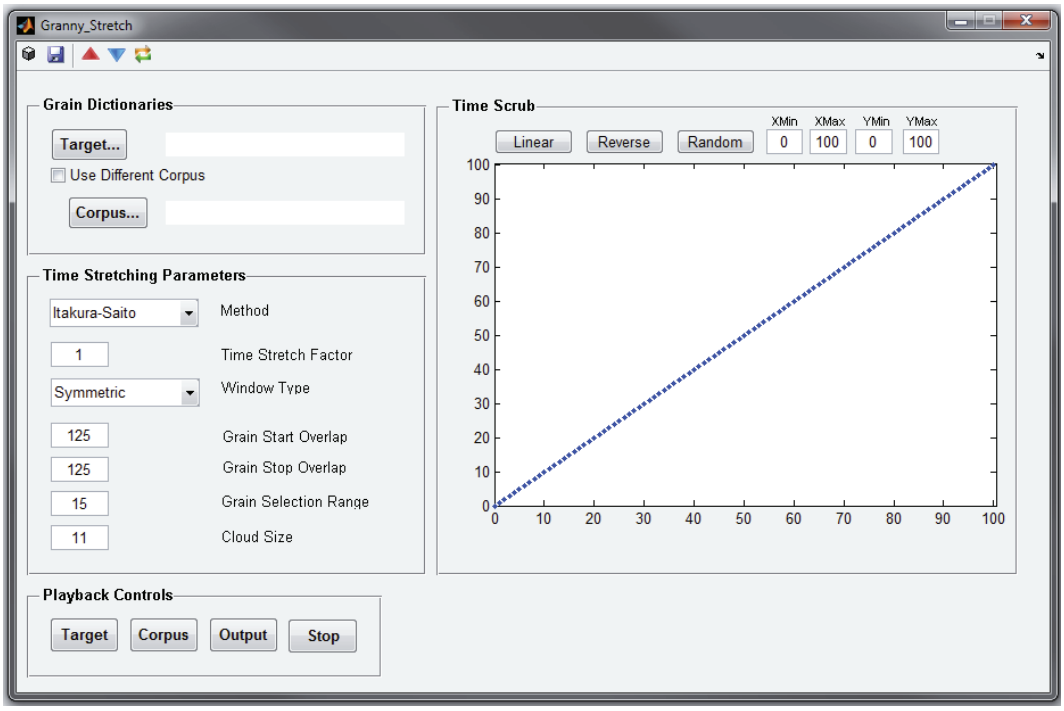

Fig. 3.5 GUI for synthesis.

eration of the granular synthesis system is to shift the time positions of grains in the corpus dictionary, making the starting time positions of the kth grain in the corpus dictionary become $i(k)$. With the time modification processes that will be explained below, synthesis with respect to the target sound could be achieved in flexible ways.

3.4.2 Time Stretching/Shrinking

One of the main time modification schemes used in the granular synthesis system is time stretching and shrinking. Once grains in the given corpus are rearranged with respect to $i(k)$, the time modification, either stretching or shrinking, is conducted by controlling intervals between $i(k)$. The time stretch factor α ($\alpha > 0$) controls the time modification. The new sample index of the starting time of grains in the corpus dictionary after time modification is given

Time Stretching Parameters	
Method	Grain Extrapolation Itakura-Saito MFCC
Time Stretch Factor	α (> 0)
Window Type	Symmetric One-sided
Grain Start Overlap	Number of samples overlapped
Grain Stop Overlap	Number of samples overlapped
Grain Selection range	Number of grains
Cloud Size	Number of grains

Table 3.2 Time stretching parameters.

as

$$i'(k) = \text{round}(\alpha \cdot i(k)) \tag{3.12}$$

$$0 < \alpha < 1 : \text{shrinking}$$

$$\alpha > 1 : \text{stretching}$$

where 'round' denotes the rounding operation. Time modification gives rise to unnecessary gaps between grains if

$$i'(k+1) - i'(k) > l_k \tag{3.13}$$

where l_k is the length of the kth grain in the corpus dictionary. Note that not only time stretching but also time shrinking could possibly create gaps since there is a chance that the length of the grain $l(k)$ from the corpus dictionary happens to be shorter than the interval $i'(k+1) - i'(k)$ after time shrinking. Fig. 3.6 shows how time modification creates gaps. These gaps give rise to audible artifacts associated with signal discontinuities. If the desired sound should be perceptually continuous, it is essential to devise a way to fill gaps

to remove the audible artifacts. In the present granular synthesis system, two different approaches for gap filling are proposed.

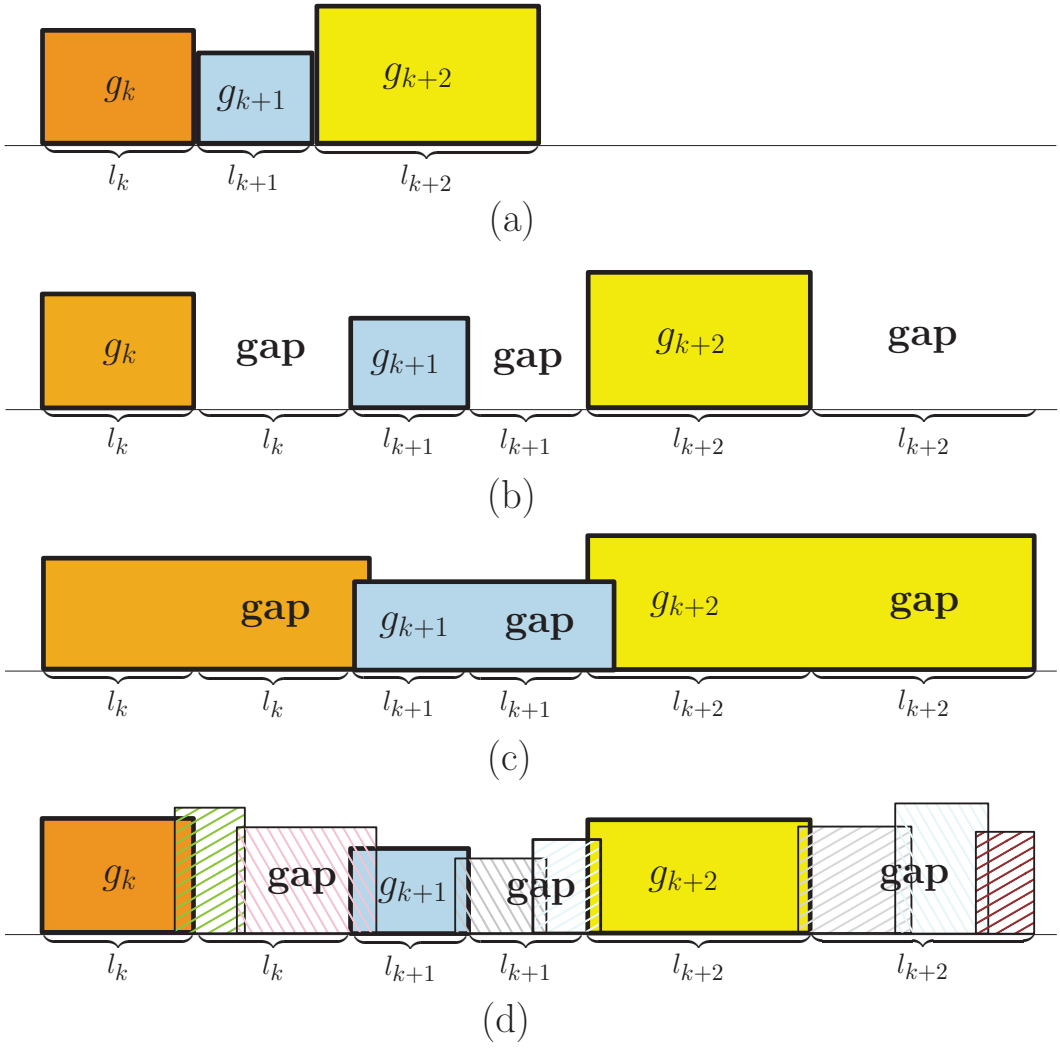

Fig. 3.6 Time stretching and gap filling. (a) original sequence of grains g_k, g_{k+1}, g_{k+2} of length l_k, l_{k+1}, l_{k+2}, respectively. (b) time stretched with the time stretch factor $\alpha = 2$. (c) Gap filling with grain extension (d) Gap filling with additional grains.

3.4.3 Gap Filling Strategies

Gap Filling with Grain Extension Method

One way to fill a gap is to extend the grain placed right before a gap. The idea of grain extension is inspired by audio signal interpolation/extrapolation techniques that have been studied and developed for application to signal restoration for disturbed and missing data [46] [47] [48]. The grain extension algorithm used here is based on LP, using samples at the end of the grain to be extended as initial data for LP.

In LP a signal sample $x(n)$ is assumed to be modeled as $\hat{x}(n)$, a linear combination of preceding samples as follows,

$$\hat{x}(n) = \sum_{k=1}^{p} a_k x(n-k) \tag{3.14}$$

where p is the order of the LP and a_k, $(k = 1, \cdots, p)$ are the LP coefficients. The error between the modeled value $\hat{x}(n)$ and $x(n)$, $e(n)$ is

$$e(n) = x(n) - \hat{x}(n) = x(n) - \sum_{k=1}^{p} a_k x(n-k) \tag{3.15}$$

The LP coefficients a_k are estimated by minimizing the error $e(n)$, and many ways of estimating the LP coefficients have been researched [27]. Generally, iterative algorithms are favored for this task because of the fast computation. One such algorithm is the Burg algorithm [27], which makes use of both forward and backward prediction errors to increase the accuracy of estimation. In [46], an audio extrapolation technique based on the Burg algorithm-based LP is proposed, which has been adopted for this grain extension. Estimated LP coefficients allow for extrapolation of a grain in such a way that past samples

are filtered with the FIR filter whose coefficients are the LP coefficients a_k.

In order first to estimate the LP coefficients to use for grain extension, we begin with a linear prediction of the last sample of a grain $g(n)$ of length L,

$$\hat{g}(L) = \sum_{m=1}^{p} a_m g(L-m) \tag{3.16}$$

where a_m are the LP coefficients estimated using the last p samples of g, and $\hat{g}(L)$ is the estimate of the last sample of the grain $g(L)$. p is the LP order. Given $g(L)$, a_m are estimated using Burg's method. Once the LP coefficients are estimated, they are used to extrapolate $g(n)$ by predicting the future samples. The first extrapolated sample, $\hat{g}(L+1)$ is obtained as

$$\hat{g}(L+1) = \sum_{m=1}^{p} a_m g(L-m+1) = \mathbf{A}\mathbf{g_1} \tag{3.17}$$

where $\mathbf{A}$ and $\mathbf{g_1}$ are

$$\mathbf{A} = [\, a_1 \;\; a_2 \;\; \cdots \;\; a_{p-1} \;\; a_p \,] \tag{3.18}$$

$$\mathbf{g_1} = [\, g(L) \;\; g(L-1) \;\; \cdots \;\; g(L-p+2) \;\; g(L-p+1) \,]^T \tag{3.19}$$

In the same way, we can proceed to produce further samples just by updating $\mathbf{g_1}$ with newly extrapolated samples. For example, in order to produce $\hat{g}(L+r)$, which is calculated as $\hat{g}(L+r) = \mathbf{A}\mathbf{g_r}$, $\mathbf{g_r}$ should be given as

$$\mathbf{g_r} = \begin{cases} [\, \hat{g}(L+r-1) \;\; \hat{g}(L+r-2) \;\; \cdots \\ \qquad \hat{g}(L+r-p+1) \;\; \hat{g}(L+r-p) \,]^T & \text{if } r > p \\ [\, \underbrace{\hat{g}(L+r-1) \cdots \hat{g}(L+1)}_{r-1} \;\; \underbrace{g(L) \;\; \cdots \;\; g(L-p+r)}_{p-(r-1)} \,]^T & \text{otherwise.} \end{cases} \tag{3.20}$$

The number of samples to be extrapolated is determined by the sum of the length of the gap and the parameter 'Grain Stop Overlap' (Table 3.2), which specifies how many samples are overlapped between neighboring grains. The LP order p can be arbitrarily chosen as long as $L > p$. In general, the more samples used to estimate the LP coefficients, the more accurate the estimate of the LP coefficients becomes [27]. Figure 3.7 gives an example of grain extension.

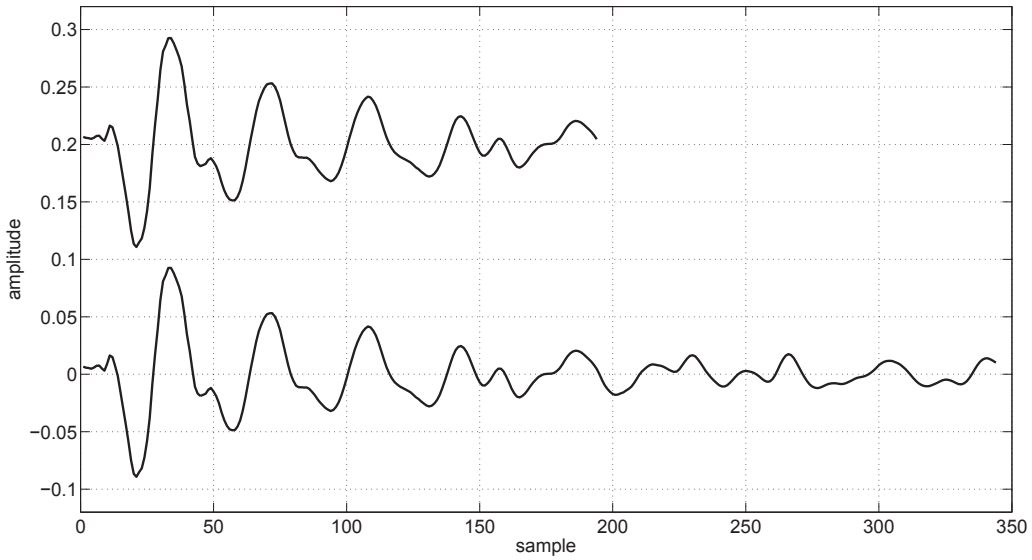

Fig. 3.7 Example of grain extension.

Gap Filling with Additional Grain-Based Method

Since the gap part would likely be similar to the parts where grains are present, natural gap filling could be achieved by placing the most similar grains into the gap. The optimal grains for the gap are determined on the basis of how similar they are to the grain placed just before a gap. Rather than extrapolating existing grains to fill gaps, these optimal additional grains are chosen from the grain dictionary and placed in the gap. This strategy would give a differ-

ent kind of audible sensation to the listeners. In order to preserve the natural perception when filling gaps in this way, it is essential to choose grains appropriately. To keep the feeling of continuity with neighboring grains, additional grains that are to be filled into gaps are selected according to the similarity to the existing grain. As the measures for representing the similarity, we use two features that are based on the spectral distance.

One is the Itakura-Saito (IS) divergence [49]. The IS divergence is a measure of the perceptual difference between two spectra, defined as follows,

$$D_{IS}(k,k') = \frac{1}{2\pi}\left[\int_{-\pi}^{\pi} \frac{P_k(\omega)}{P_{k'}(\omega)} - log\frac{P_k(\omega)}{P_{k'}(\omega)} - 1\right] d\omega \tag{3.21}$$

where $P_k(\omega), P_{k'}(\omega)$ are the two spectra to be compared. The other is Mel Frequency Cepstral Coefficients (MFCC). The MFCC are a perceptually based spectral feature widely used in speech recognition and music information retrieval.

In order to obtain the MFCC, the magnitudes of DFT coefficients $|X_k|$ of signal $x(n)$ are first scaled in frequency so that the frequency is transformed to log scale using the Mel filter bank $H(k,m)$ in such a way that

$$X'(m) = ln(\sum_{k=0}^{N-1} |X(k)| \cdot H(k,m)), \qquad m = 1, 2, \cdots, (M << N). \tag{3.22}$$

where M and N are the number of filter banks and the DFT length, respectively. $H(k,m)$, the mth band filter, has a triangular shape, defined as,

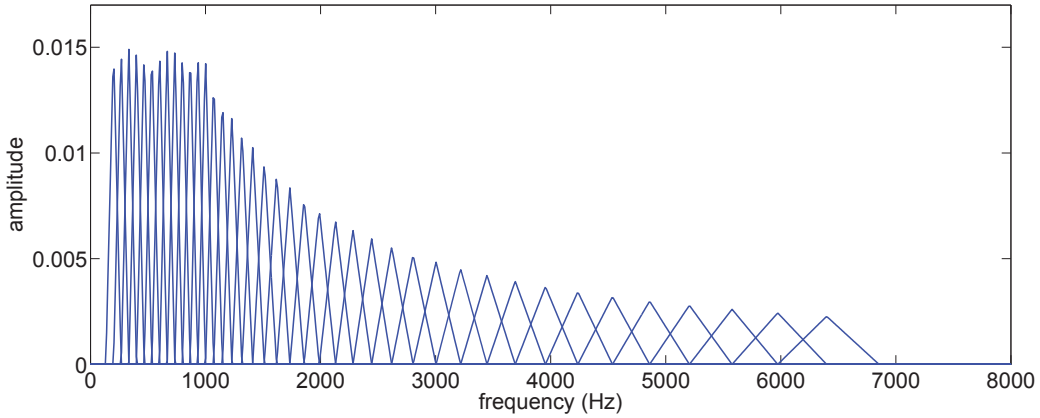

Fig. 3.8 Triangular mel-scale filter bank from Auditory toolbox [1].

$$H(k,m) = \begin{cases} \dfrac{k - k_{c,m-1}}{k_{c,m} - k_{c,m-1}} & \text{if } k_{c,m-1} \leq k < k_{c,m} \\ \dfrac{k_{c,m+1} - k}{k_{c,m+1} - k_{c,m}} & \text{if } k_{c,m} \leq k < k_{c,m+1} \\ 0 & \text{otherwise,} \end{cases} \tag{3.23}$$

where $k_{c,m}$ is the frequency bin number corresponding to the center frequency of the mth band $f_{c,m}$ in Hz as $k_{c,m} = N f_{c,m}/f_s$, where f_s is the sampling frequency. The frequency in Hz is mapped onto the Mel scale according to the formula [50]:

$$\phi = 2565 \ \log_{10}\left(\frac{f}{700} + 1\right) \tag{3.24}$$

The center frequency of the mth band on the Mel scale is given as

$$\phi_{c,m} = m\frac{\phi_{max} - \phi_{min}}{M+1} \tag{3.25}$$

where ϕ_{max}, ϕ_{min} are the Mel frequency scales of the upper bound frequency f_{max} and the lower bound frequency f_{min}, derived using Eq. 3.24. The MFCC,

$c(l)$, are obtained by conducting the discrete cosine transform (DCT) of $X'(m)$ [50].

$$c(l) = \sum_{m=1}^{M} X'(m) \cos\left(l\frac{\pi}{M}\left(m - \frac{1}{2}\right)\right) \tag{3.26}$$

Let $\mathbf{C}_k = [\ c(0)\ \ c(2)\ \cdots\ c(M-2)\ \ c(M-1)\ ]^T$. Then the MFCC distance between the two grains is given as

$$D_{MFCC}(k, k') = \frac{\mathbf{C}_k \cdot \mathbf{C}_{k'}}{|\mathbf{C}_k||\mathbf{C}_{k'}|} \tag{3.27}$$

Grain Selection Range and Cloud Size

The simplest way to select an additional grain would be to select the grain that has the smallest measurable distance from the preceding grain just before a gap. In principle, the methods based on additional grains are supposed to compare the target grain, the existing one already given just before a gap, with all the remaining grains in the corpus dictionary. This often requires heavy computation when the size of the dictionary is large. In particular, if the target sound is relatively homogeneous, then searching through the entire grain dictionary would be excessive in the extreme as it would be highly likely that all the grains in the dictionary are spectrally similar. In order to let users adjust the tradeoff between the computation load and the extent to which the target grain and the chosen grain are similar to each other, another parameter, referred to as the 'Grain Selection Range' (Fig. 3.5, Table 3.2), is proposed. The grain selection range determines a pool of grains in which a search for an additional grain is conducted. If the grain selection range is given n_{gsr} and the kth grain is the target grain, the candidate grains are defined by their orders

in the dictionary, k', as

$$\{k - n_{gsr} \leq k' \leq k + n_{gsr},\ k' \neq k\} \tag{3.28}$$

The number of grains in the range are then $2n_{gsr} - 1$, excluding the target grain itself.

Another issue that can arise in the additional grain scheme is the repetition of the same grain, particularly when more grains than one are needed to fill the gap. It is highly likely that the grain that has been chosen once will be chosen again, especially when the grains in the grain selection range are alike in terms of spectral content. Owing to this chance of multiple-selection of the same grain, audible artifacts could often be found in the resulting synthesized sound. To prevent this, instead of strictly selecting the very best matched grain, that is, the one that has the smallest distance from the target grain, an additional grain is randomly chosen from among a pool of best-matched grains. The number of grains in this pool is referred to as the 'Cloud Size' (Fig. 3.5, Table 3.2). The larger the cloud size, the more random the selection. If the cloud size is set to 1, then the best-matched grain is selected. Note that if the cloud size is given as n_{cs}, then it should satisfy the condition $n_{cs} \leq n_{gsr}$. Once an additional grain is selected, the amplitude of that grain is normalized to the average power of the target grain preceding the gap and the grain succeeding the gap.

3.4.4 Windowing

As grains are overlapped and added, it is likely that audible artifacts occur at the joints of grains as a result of abrupt change of amplitude. To remedy this situation, a grain is first weighted with a window function to taper either

the end side or the beginning side, or both sides. The shape of the window is determined by the length of a grain and the values of the 'Grain Start Overlap' (Fig. 3.5, Table 3.2) and the 'Grain Stop Overlap' parameters. The Grain Start Overlap is the number of samples at the beginning of the grain to be tapered, and the Grain Stop Overlap is the number of samples at the end of the grain to be tapered, respectively. Let n_{start}, n_{stop} be the values of the Grain Start Overlap and the Grain Stop Overlap and L be the length of the grain in concern, then using the Hann window, the window is defined as

$$w(n) = \begin{cases} 0.5\left(1 - \cos\left(\pi \dfrac{n}{n_{start}}\right)\right) & , \quad \text{for} \quad 0 \leq n \leq n_{start} \\ 1 & , \quad \text{for} \quad n_{start} < n < L - n_{stop} \\ 0.5\left(1 + cos\left(\pi \dfrac{n - (L - n_{stop})}{n_{stop}}\right)\right) & , \quad \text{for} \quad L - n_{stop} \leq n \leq L \end{cases} \tag{3.29}$$

Depending on values of n_{start} and n_{stop}, one can make a window either double-sided or one-sided. In general, a one-sided window with no tapering at the beginning is used to preserve the attack transient of a grain. This is often the case when using the grain extension method. On the other hand, a double-sided window could be used to smooth both sides of a grain used for bridging two grains into the gap when using the additional grain method. Figure 3.9 shows an example of a window applied to a grain.

3.4.5 Grain Extension Method vs. Additional Grain-Based Method

One thing to take note of is the characteristics of synthesized sounds in accordance with the proposed gap-filling methods. The principal difference of the grain extension method and the additional grain-based method is the grain

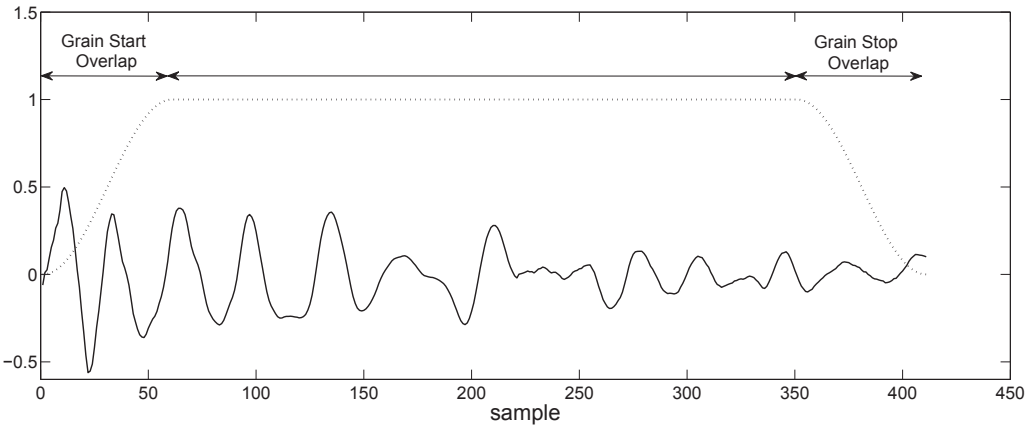

Fig. 3.9 Window used for gap filling. 'Grain Start Overlap' and 'Grain Stop Overlap' and the length of the grain determine the overall length of the window.

density after synthesis. The grain density of the sound synthesized with the grain extension method varies proportionally with the time stretch factor, whereas that of the sound synthesized with the additional grain-based method is invariant with respect to the time stretch factor. Figure 3.10 shows an example of synthesis with time stretching. Depending on the nature of the given sound and the user's purpose, either method could be preferred. For example, one can create two different kinds of clap sounds based on time modification.

Figure 3.11 shows the difference of the time stretched synthesis due to the choice of gap-filling methods[1]. The synthesis using the grain extension method generally results in a decrease of the grain density inversely proportional to the time stretch factor; on the other hand the synthesis based on the additional grain-based method keeps the grain density of the original clap sound. However, keeping grain density does not necessarily mean keeping the rhythmic nature of the original sound not only because perception of rhythm has to do with the spectral nature of the grains but also because the lengths of grains could be very different from one another. This aspect actually provides

[1]Sound examples are available at `http : www.music.mcgill.ca/ ∼ lee/thesis/granular`

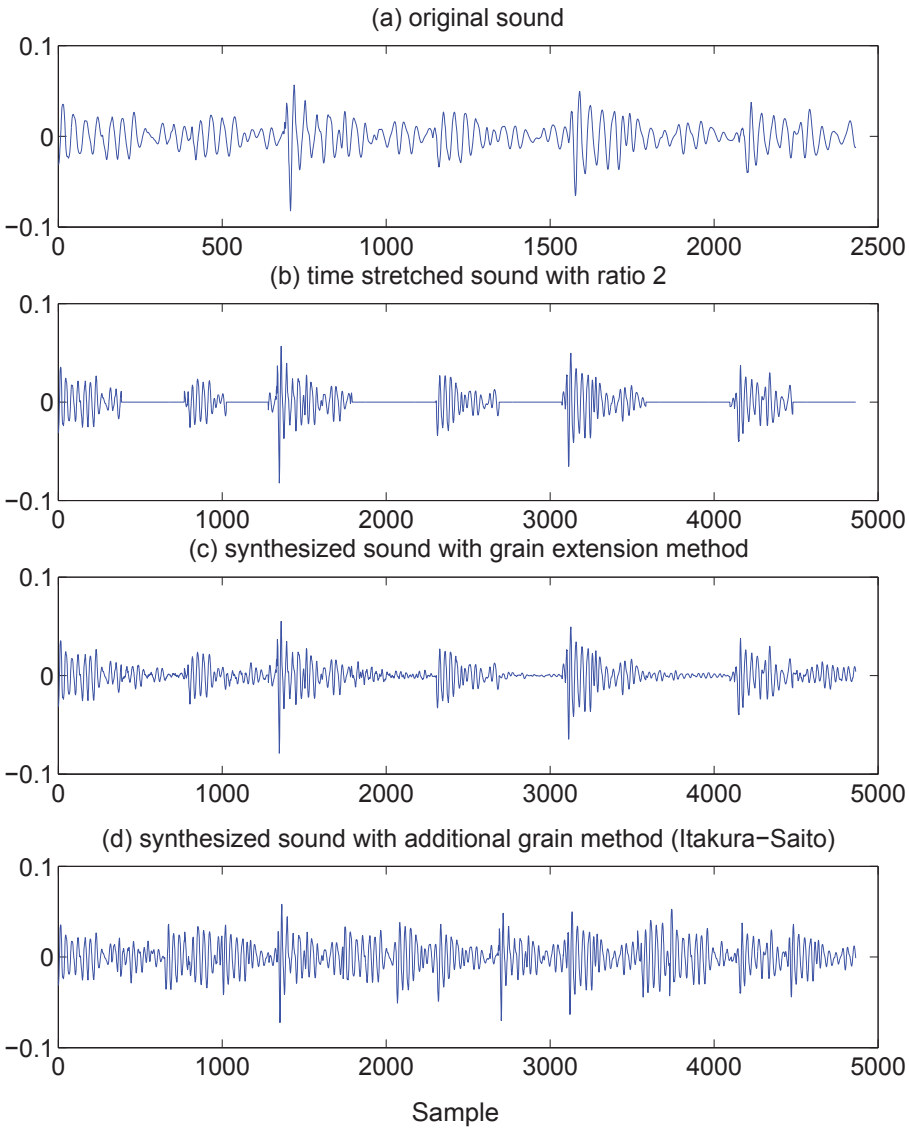

Fig. 3.10 (a) Original sound. (b) Original sound stretched with the time stretch factor $\alpha = 2$. (c) Gap filling with the grain extension method. (d) Gap filling with the additional grain method.

users with another option in synthesis, allowing for synthesized sounds that are sparse in terms of the gain density. In this case, the grain extension method plays the role of polishing each grain to avoid incurring audible artifacts due to the abrupt ends of grains.

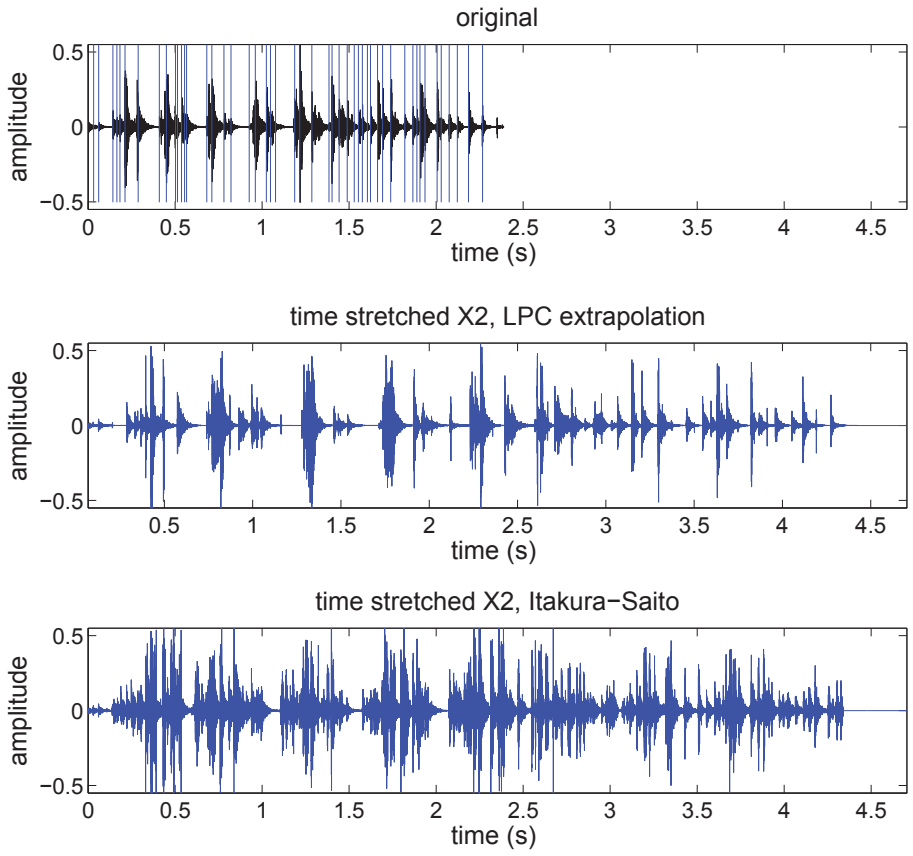

Fig. 3.11 Time stretched clap sounds. a) Original sound. Blue vertical bars denote the grain boundaries. (b) Time stretched sound by a factor $\alpha = 2$, with the grain extension method. (c) Time stretched sound by a factor $\alpha = 2$, with the additional grain-based method (Itakura-Saito).

3.4.6 Grain Time Remapping

The concept of grain time remapping allows for more variations. Since all the grains have their own time positions representing locations of grains on the time axis, grain time remapping often allows for creating different scenarios

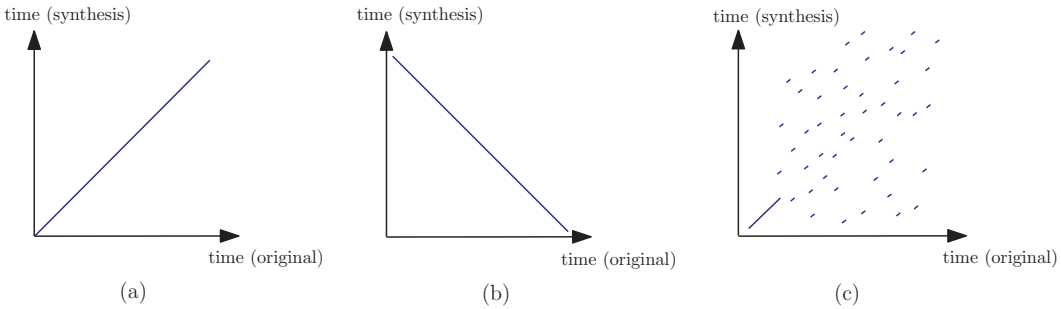

Fig. 3.12 grain time remapping examples. (a) no grain time remapping. (b) grain time remapping in reverse order. (c) random grain time remapping.

for sound generations in the same environment. This can result in many interesting effects. For example, grain time remapping of the rolling ball sound would provide listeners with a variety of acoustic sensations since the time sequence of the grains has to do with the trajectory of the rolling ball, as mentioned in the previous chapter. Thus adjusting the time sequence could actually have the effect of changing the trajectory of the ball. For example, if the time sequence of the grains is reversed (Fig. 3.12-(b)), the resulting synthesis will sound as if the ball were rolling backward along the trajectory of the original sound.

3.5 Discussion

The outcome of the current research consists of two components. One is granular analysis and the other is granular synthesis. Both components were implemented in MATLAB and managed through GUIs. The granular analysis system is designed to detect onset-like events so that it can segment a given sound into grains. The granular analysis system is also able to discern stationary/non-stationary regions in the given sound and apply different segmentation parameters for each region, which enables users to apply different

criteria for defining the grain in each region. In addition, useful audio features such as Energy, Spectral Centroid, Spectral Tilt and Spectral Flatness are calculated for each grain for potential use in synthesis. Segmented grains are tagged with timing information and audio-features are stored in a dictionary.

The research has also developed a novel granular synthesis system whereby a user can synthesize a sound in conjunction with the granular analysis system in flexible ways. The user can modify the temporal aspect of the sound in various ways. Not only conventional time-scaling (stretching/shrinking) but also user-defined grain time remapping of grains with convincing sound quality is available in this synthesis system.

Because it is based on the granular analysis/synthesis framework, the proposed scheme is good at re-producing sounds that can be well segmented. Such sounds would be those of breaking, clapping, and rolling that take place under particular conditions, for example, rolling on a 'rough' surface. In these sounds, micro sound events are distinguishable to a reasonable extent so that segmented grains are likely to correspond to micro physical interactions among objects. With such sounds, all aspects presented in the proposed scheme will show the best performance. For example, given a rolling sound in which micro contacts between the ball and the surface are well transformed to distinct micro sound events, grain analysis will result in well-segmented grains, which in turn let the grain synthesis part re-synthesize the given sound with good quality. In addition, temporal modifications would be capable of re-synthesizing sounds as if the physical condition that governs sound generation were changed. For example, remapping the time order of grains in reverse would generate a sound as if a ball were rolling backward in the same trajectory as the original sound; this result is due to the fact that the modes are excited and suppressed,

depending on the location of excitation on the surface.

Another category of sounds that would fit the proposed scheme well is that of constant, regular, 'stationary' sounds. Examples of such sounds are a gentle fire burning, the flowing of a brook and falling rain, and so forth. Using the stationarity analysis, those sounds can be analyzed either microscopically or macroscopically. Owing to the nature of these sounds, time modification, especially stretching, can be easily done, simply by using the additional grain-based method, keeping the perceptual event density as much as possible.

The weak points of the proposed scheme are rooted in the fact that the proposed scheme is based on the granular analysis/synthesis framework. Particularly when it is hard to segment given sounds properly so resulting grains have little or no relations with micro physical interactions, re-synthesis does not produce the desired results. One representative example of this is the sound of thunder. Generally, thunder sounds are continuous, giving the overall impression of a continuum, but they also have a dynamically varying aspect. Due to the continuous nature, it is hard to carry out segmentation that would result in meaningful grains that capture distinct physical interactions. This would in turn result in undesired re-synthesized sounds especially when temporal modifications are made. Also, sounds of musical instruments would not fit properly into the proposed scheme since the characteristics associated with the sounds of musical instruments that would be critical in carrying out proper segmentation are not taken into account.

Future work will include several research tasks that could potentially enhance the current research outcomes. One would be finding a clever way for grain compression other than using the 'Offset Threshold' parameter. In general, it is likely that redundant grains exist in a dictionary, and they in-

cur unnecessary consumption of computer resources. By clustering redundant grains through the use of a proper machine learning technique, the size of a dictionary can be reduced while the quality of sound synthesis is maintained. Another problem to think about is how to figure out the inherent rhythmic aspect of a given sound. In contrast to music or speech, environmental sounds are quite often non-rhythmic or have rhythms that are hard to analyze (e.g. the sound of applause). However, if we could analyze the rhythm of a sound, it would be beneficial insofar as it would broaden the flexibility of the synthesis system.

Chapter 4

Extraction and Modeling of Pluck Excitations in Plucked-String Sounds

4.1 Introduction

In this chapter, as an attempt to explore various aspects of sources used for synthesizing musical sounds, we investigate ways of extraction and modeling of excitations associated with physical models of plucked strings. We first propose a simple but physically intuitive method to extract the excitation signal from a plucked string signal in the time domain. By observing the behavior of the traveling wave components in the given plucked string sound signal and comparing that to a digital waveguide (DW) simulation, the pluck excitation is 'visually' identified and extracted simply by time windowing. Motivated by this time-windowing-based method, another method for extracting excitations is proposed. This latter method is based on inverse-filtering given a physical model of plucked strings which can be viewed as a source-filter framework.

This method aims at extracting the excitation of the most compact form in time by taking into account the plucking position and the pickup position, and the results of both the time-windowing based method and the inverse-filtering based method are compared to show they yield almost the same results, which are compact in the time-domain and accurately reflect the profile of pluck excitations in the acceleration (force) dimension. In addition, an approach to model extracted pluck excitations using an existing glottal flow derivative model for speech processing is proposed, and the estimation of the model parameters is also discussed.

4.2 Background

Since the concept of physical modeling synthesis has emerged, plucked strings have been one of the major applications. The sound quality of plucked-string physical models is often highly dependent on the excitation model, or the way that energy is input into the system since the action of plucking is almost the only domain over which a performer has control in the real world. It is therefore essential to use proper excitations to synthesize natural plucked string sounds that successfully convey the performer's articulations in conjunction with physical model-based synthesis techniques. For the purpose of both the analysis and the synthesis of 'natural sounding' plucked string sounds, using excitations analyzed from real plucked string sounds is often desirable. Consquently, 'analyzing out' the plucking excitations has attracted the attention of researchers, leading to many interesting and useful outcomes.

The first physical model of plucked string sounds dates back to the well-known Karplus-Strong (KS) model [51] which is actually a quasi-physical model of plucked strings regarded as the predecessor of physical models of

plucked strings. A noise signal is used as a pluck excitation in the KS model. Several works [15][52][53] have addressed the extraction of the pluck excitation using more advanced physical models in the form of digital filters. These works all involve inverse filtering with recorded plucked string sound signals as a way of decomposing the source and the filter in the source-filter framework. This inverse filtering approach is also found in [54], where the single delay loop (SDL) model is employed as the physical model of a plucked string. A general source-filter model, without a specific physical model customized to the musical instrument concerned, is often used for excitation extraction, on the assumption that the mechanism of the sound generation from certain musical instruments follows the excitor-resonator relation [12]. In this approach, a resonant filter that represents harmonic peaks in the spectrum of a sound is constructed based on the analysis of harmonic peaks in the frequency domain and then inverse filtered with the target sound to extract the excitation. In [16], the authors propose a method to extract the pluck excitation in a non-parametric way by suppressing all the peaks in the magnitude response of a plucked string sound and replacing those suppressed spectral magnitudes with the values realized from the probability distribution associated with the amplitudes of neighboring magnitudes.

In regards to the modeling of pluck excitations, there has been little research compared to the extraction of excitations. One of the first attempts to model a pluck excitation can be found in [55], where the extended KS model is proposed. A one-pole filter is used to implement the difference of the up picking and down picking, which is one of the most fundamental articulations in playing plucking-based string instruments. In [56], Smith and Van Duyne modeled the excitation signal fed into the physical model of the piano. They

paid attention to the behavior of a hammer-string action and modeled the shape of the initial force pulse wave created from the hammer strike, the reflected pulse returning to the strike point, and the subsequent pulse generated from the interaction between the reflected pulse and the hammer still in contact with the string, using a few low-pass filtered impulses. In [54], the authors addressed the modeling of pluck excitations using principle component analysis (PCA). They first extracted pluck excitations using the SDL-based inverse filtering, as mentioned above and then applied the PCA to the collected pluck excitations and built a 'codebook' of pluck excitations that could be used for synthesis. Another interesting approach for modeling pluck excitations can be found in [53] and [57], where the FIR filter is used to describe the shape of pluck excitations. This approach might look similar to the methods proposed in [55] and [56] inasmuch as the excitations are modeled using simple FIR filters; however, the methods in [53] and [57] are different in that the filter parameters are set based on the estimation of the actual excitation signal data.

4.3 Digital Waveguide Theory of Ideal Vibrating String

In this section, we review the digital waveguide theory, particularly as it relates to wave propagation on an ideal string. The ideal string is assumed to be perfectly flexible and elastic.

4.3.1 One-dimensional Digital Waveguide Theory

The wave equation for wave propagation on the ideal string is

$$K\frac{\partial^2 y}{\partial x^2} = \epsilon\frac{\partial^2 y}{\partial t^2} \tag{4.1}$$

where K, ϵ, y are the string tension, the linear mass density of the string, and the displacement of the string, respectively. A solution to Eq. 4.1, attributed to d'Alembert, is [58]

$$y(t, x) = y_r(t - x/c) + y_l(t + x/c) \tag{4.2}$$

where $y_r(\cdot)$ and $y_l(\cdot)$ are functions that describe a preset string shape, traveling to the right and to the left at the speed of c, as defined in Eq. 4.3, respectively.

$$c = \sqrt{\frac{K}{\epsilon}} \tag{4.3}$$

In the digital waveguide, the right-going and the left-going displacement com-

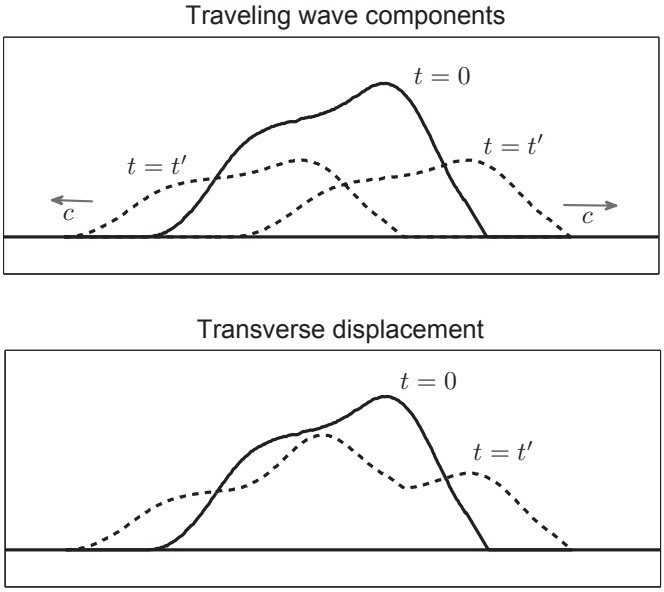

Fig. 4.1 Traveling wave components and the transverse displacement. The waveforms shown in the top pane are the right-going and the left-going traveling wave components at time $t = 0$ and $t = t'$. The waveforms shown in the bottom pane are the transverse displacements at time $t = 0$ and $t = t'$, sums of the two traveling wave components shown in the top pane.

ponents in the traveling wave solution of Eq. 4.2 are sampled at sampling in-

terval T (in seconds), which relates to the sampling frequency f_s (in Hz) as $T = 1/f_s$. Accordingly, the spatial variable x is sampled at the interval of X, given as $X = cT$, which can be interpreted as the distance the traveling waves move over time T. By sampling in both time and space, continuous temporal and spatial variables can be changed so that

$$x \longrightarrow x_m = mX \tag{4.4}$$

$$t \longrightarrow t_n = nT \tag{4.5}$$

Then, Eq.(4.2) can be written as below, using the new variables:

$$y(t_n, x_m) = y_r(t_n - x_m/c) + y_l(t_n + x_m/c) \tag{4.6}$$

$$= y_r(nT - mX/c) + y_l(nT + mX/c) \tag{4.7}$$

$$= y_r[(n-m)T) + y_l[(n+m)T]) \tag{4.8}$$

By defining the new notations as below, we can omit T in Eq. 4.8

$$y^+(n) = y_r(nT), \qquad y^-(n) = y_l(nT) \tag{4.9}$$

The superscripts '+' and '−' denote the traveling directions, to the right and left, respectively. The resulting expression for the transverse displacement at time n and location m is given as the sum of the two traveling wave components:

$$y(t_n, x_m) = y^+(n-m) + y^-(n+m). \tag{4.10}$$

In the digital waveguide, due to the definition of $X = cT$, traveling wave components can be described with one type of variable (interpreted either as temporal or spatial) with delays. The right-going traveling wave term $y^+(n - m)$ can be interpreted as the output of the m-sample delay line with the input $y^+(n)$ and, similarly, the left-going traveling wave term $y^-(n + m)$ with the input $y^-(n)$. As can be seen in Fig. 4.2 which illustrates a part of the digital waveguide representing the traveling-wave solution of Eq. 4.10, the physical transverse displacement at any spatial position at time n can be computed simply by adding the outputs of the upper delay line, which shifts samples to the right, and of the lower delay line, which shifts samples to the left, at the desired spatial position. In Fig. 4.2, two outputs at $x = 0$ and $x = 3X$ are shown as examples.

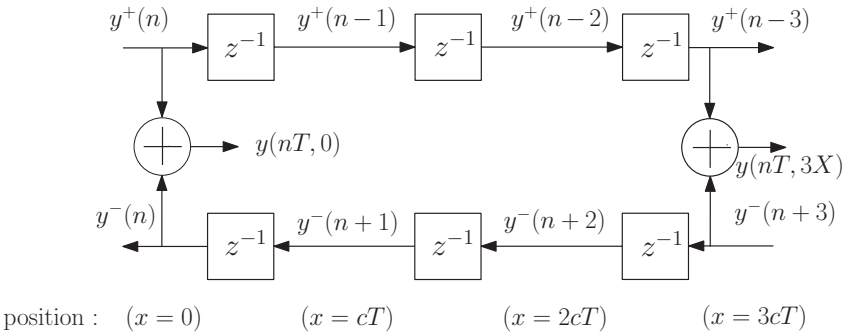

Fig. 4.2 DWG simulation of the ideal, lossless waveguide after [2]

So far the transverse displacement y has been used as a wave variable to describe the digital waveguide model. In the digital waveguide, propagation of other physical quantities, such as velocity and acceleration, could also be described as long as they can be depicted as wave equations.

4.3.2 Ideal Digital Waveguide Plucked String Model

In this section, the digital waveguide model of the ideal plucked string is reviewed. The ideal plucked string can be characterized simply with an initial string displacement shape $y(0, x)$ and a zero initial velocity distribution along the string $\dot{y}(0, x)$. This can be interpreted as releasing a string which has been pulled from its rest position. Figure. 4.3(a) shows an example of an initial condition. The length of the string is L meters and accordingly the delay length of the 'string loop', the total number of delays in both delay lines, is defined as $N_{loop} = 2L/X$. Another simple choice for describing the

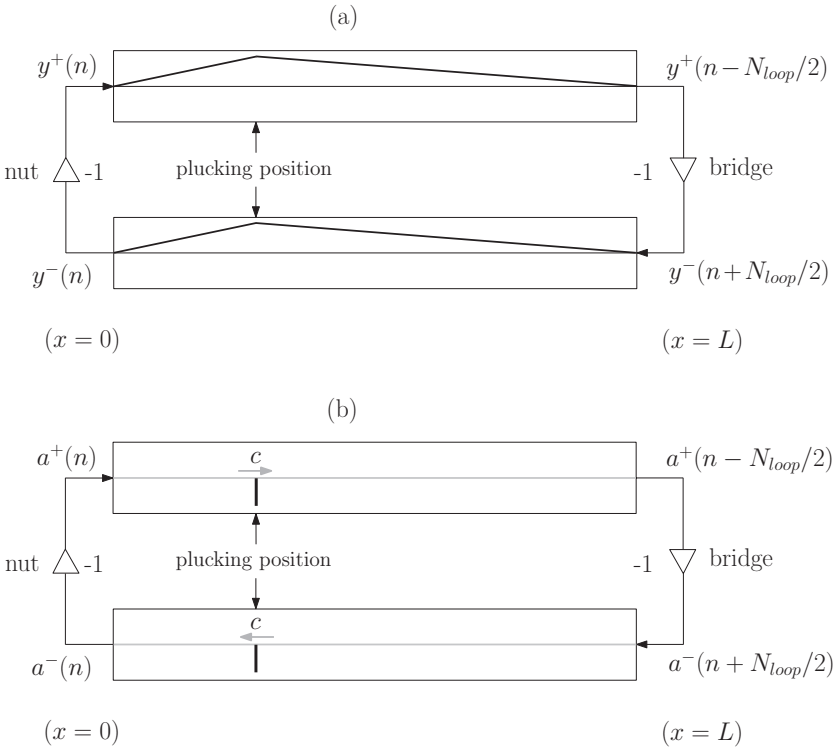

Fig. 4.3 Ideal plucked string digital waveguide models. (a) A simulation using wave variables of displacement $y(n)$. The initial condition $y(0, x)$ is characterized by the shapes loaded in the delay lines. (b) A simulation using wave variables of acceleration $a(n)$. The initial condition $a(0, x)$ is characterized by the impulses loaded in the delay lines. All figures are after [2].

behavior of a plucked string in the digital waveguide is to use acceleration wave

variables $a(t,x)$, which is $\ddot{y}(t,x)$, since feeding impulses into both delay lines of a digital waveguide corresponds to the ideal pluck. Figure. 4.3(b) shows an initial acceleration distribution along the string where impulses are loaded at the plucking point. The advantage of using an acceleration wave variable is that the output of the digital waveguide is actually the impulse response system, so we can use the LTI system theory for further investigation.

We also assume rigid terminations as the boundary condition applied to both ends of a string as it is the simplest termination scenario that would be useful for describing the behavior of the ideal plucked string. As both displacements and accelerations should be 0 at both ends, which are assumed to be rigid terminations, we have

$$y(t,0) = 0, \qquad y(t,L) = 0 \tag{4.11}$$

$$a(t,0) = 0, \qquad a(t,L) = 0 \tag{4.12}$$

Therefore, traveling wave components in the digital waveguide at the terminations should satisfy

$$y^+(n) = -y^-(n) \tag{4.13}$$

$$y^-(n + N_{loop}/2) = -y^+(n - N_{loop}/2) \tag{4.14}$$

$$a^+(n) = -a^-(n) \tag{4.15}$$

$$a^-(n + N_{loop}/2) = -a^+(n - N_{loop}/2) \tag{4.16}$$

Therefore, a reflection coefficient of -1 should be placed at both ends of the digital waveguide model as seen in Fig. 4.3(b).

4.4 Time Domain Profile of the Plucked String

In order to observe the motion of a string when it is plucked, a string of an electric guitar is plucked and the signal is recorded using a standard audio interface. A general guitar cable with quarter inch jacks is used to connect the guitar with the audio interface. As both terminations of a string of the electric guitar are almost ideally rigid, especially compared to an acoustic guitar, a signal captured by the electromagnetic pickup mounted on the electric guitar is preferred to a signal generated by plucking a string attached to an acoustic guitar and recorded by a microphone.

The guitar used for recording is a Fender Stratocaster American Standard model. The scale length (the distance from the nut to the bridge) L_{sc} equals 64.8 cm [59][1]. Figure 4.4 shows an example of an electric guitar signal recorded as previously specified. For this signal, the lowest E string is plucked with a plectrum at the middle of the string (the 12th fret) and the front pickup (the one closest to the nut) is chosen to capture the string vibration. The distance between the bridge and the front pickup is 16 cm and the sampling frequency f_s is set to 44100 Hz. The fundamental frequency f_0 is 83 Hz (obtained by simply taking a look at the spectrum and the autocorrelation of the signal).

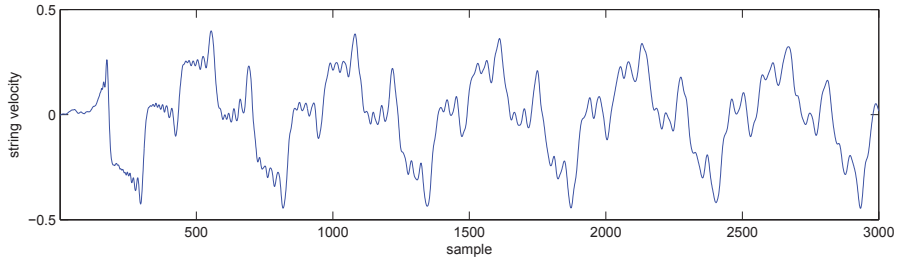

Fig. 4.4 Beginning of a plucked string sound observed through electromagnetic pickup.

Given the values describing the strings vibration, the digital waveguide

[1] `http://www.fender.com/products/americanstandard/models.php?prodNo=011040`

simulation of a string vibration is carried out for comparison with the recorded signal (see Fig. 4.5). The length of the delay line N, half of the 'string loop' length, the pickup position N_{pu}, a number of delays between the bridge and the pickup, and the plucking position N_{pl} are derived as

$$N = \frac{f_s}{2f_0} = 265.6627 \approx 266 \tag{4.17}$$

$$N_{pu} = N - \frac{16 \times 266}{64.8} = 200.3210 \approx 200 \tag{4.18}$$

$$N_{pl} = N/2 \approx 133 \tag{4.19}$$

For simplicity, we have assumed rigid terminations, considering neither the effects of the bridge and the nut nor the frequency-dependent attenuation. The filters $P(z)$ and $I(z)$ in Fig. 4.4 account for the effect of an electromagnetic pickup normally used for an electric guitar. $P(z)$ is the transfer function that represents the characteristic of the pickup, we assumed $P(z) = 1$ for simplicity here. $I(z)$ converts between wave variable types, which is necessary because of the nature of a pickup. As a pickup mounted on an electric guitar measures the induced electromotive force (EMF) caused by the change of magnetic flux through the coil in the pickup, according to Faraday's law of induction [60] the EMF ε is given as

$$\varepsilon \propto -\frac{d\Phi_B}{dt} \propto -\frac{dy}{dt} \tag{4.20}$$

where Φ_B is the magnetic flux through the coil. As the change of Φ_B is proportional to the change of the string displacement just above the pickup coil, what the pickup measures is proportional to the time derivative of the string displacement, or the string's velocity. Thus $I(z)$ should be an integrator as the wave variable we use is acceleration. We have used a simple leaky

integrator for $I(z)$ [2] given as below,

$$I(z) = \frac{1}{1 - gz^{-1}} \tag{4.21}$$

where g is a loss factor slightly less than 1. Hence, the final output of the digital waveguide model will be $v_{N_{pu}}(n)$, the velocity of the string.

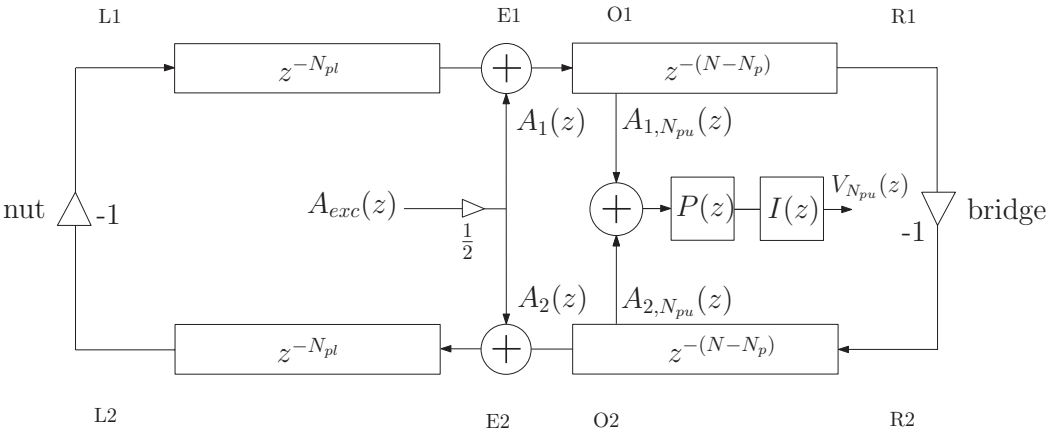

Fig. 4.5 Acceleration wave variable-based digital waveguide plucked string model with rigid terminations.

4.4.1 Excitation Extraction by Time Windowing Method

In order to obtain the impulse response of the DW system, input acceleration $a_{exc}(n)$, the inverse z-transform of $A_{exc}(z)$ in Fig. 4.5, is set to -1. $a_{exc}(n)$ is then equally split and fed into each delay line as shown in Fig. 4.5. $A_{N_{pu}}(z)$ in Fig. 4.5 is the acceleration at the pickup position, which is given as the sum of traveling wave components from the upper delay line, and the lower delay line at N_{pu}, $A_{N_{pu}}(z) = A_{1,N_{pu}}(z) + A_{2,N_{pu}}(z)$, before going into the pickup transfer function $P(z)$. Figure 4.6 shows $a_{N_{pu}}(n)$, the inverse z-transform of $A_{N_{pu}}(z)$. Thus the impulse response of the overall system $v_{N_{pu}}(n)$, the inverse z-transform of $V_{N_{pu}}(z)$, can be obtained by simply integrating $a_{N_{pu}}(n)$ as illustrated in Fig. 4.7.

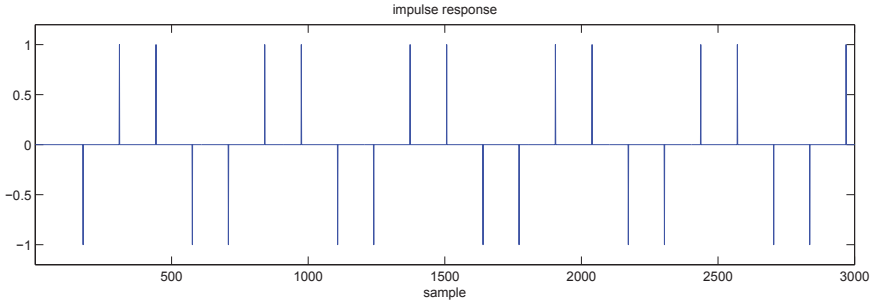

Fig. 4.6 The impulse response of the DW string model $a_{N_{pu}}(n)$ in acceleration prior to entering the pickup.

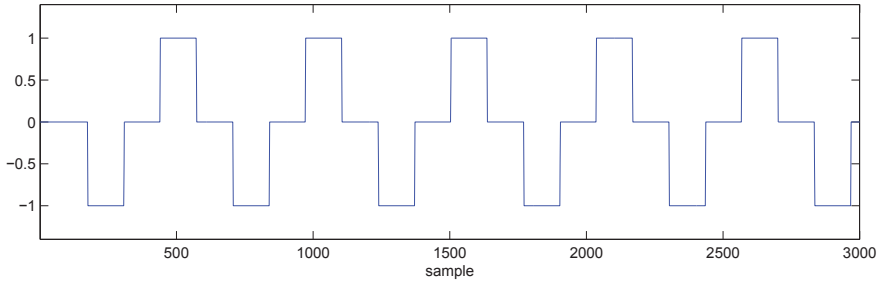

Fig. 4.7 The impulse response of the DW string model $v_{N_{pu}}(n)$ obtained by integrating $a_{N_{pu}}(n)$.

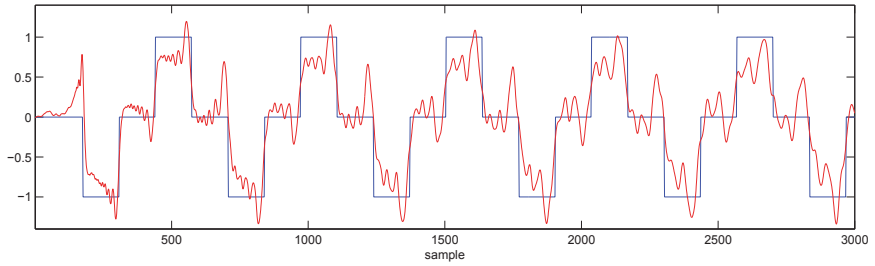

Fig. 4.8 Recorded signal $y(n)$ and the impulse response $v_{N_{pu}}(n)$.

In Fig. 4.8, the impulse response of the DW model $v_{Npu}(n)$ and the recorded signal $y(n)$ are depicted together for the sake of comparison. We can note the similarity in the time evolution of the 'bump' patterns in both $y(n)$ and $v_{Npu}(n)$. The phases of bumps in $y(n)$ and $v_{Npu}(n)$ vary in the same manner. However, the signal $y(n)$ that we actually record represents velocity waves because the pickup functions as an integrator. Thus, considering that the excitation we wish to extract from $y(n)$ is acceleration, we need to differentiate

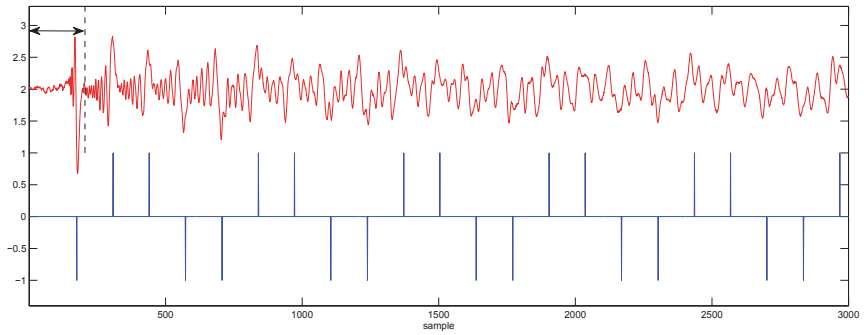

Fig. 4.9 Top: differentiated recorded signal $y'(n)$. The portion under the arrow is $\tilde{a}_{exc}(n)$. Bottom: $a_{L_p}(n)$

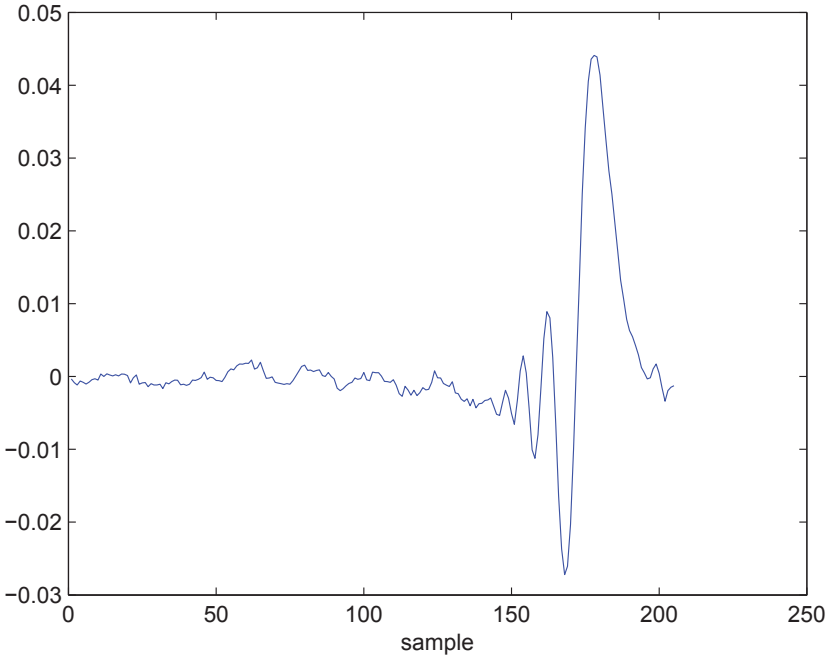

Fig. 4.10 $\tilde{a}_{exc}(n)$

$y(n)$ to $y'(n)$ for comparison to $a_{N_{pu}}(n)$. Figure 4.9 shows $y'(n)$ and $a_{N_{pu}}(n)$ together. The signal phase, or reflection, characteristics of $y'(n)$ and $a_{N_{pu}}(n)$ vary in a similar manner over time. This suggests that the excitation signal actually travels on the string in the same way that the impulse does in the DW simulation. Therefore, by carefully taking a look at both signals, we

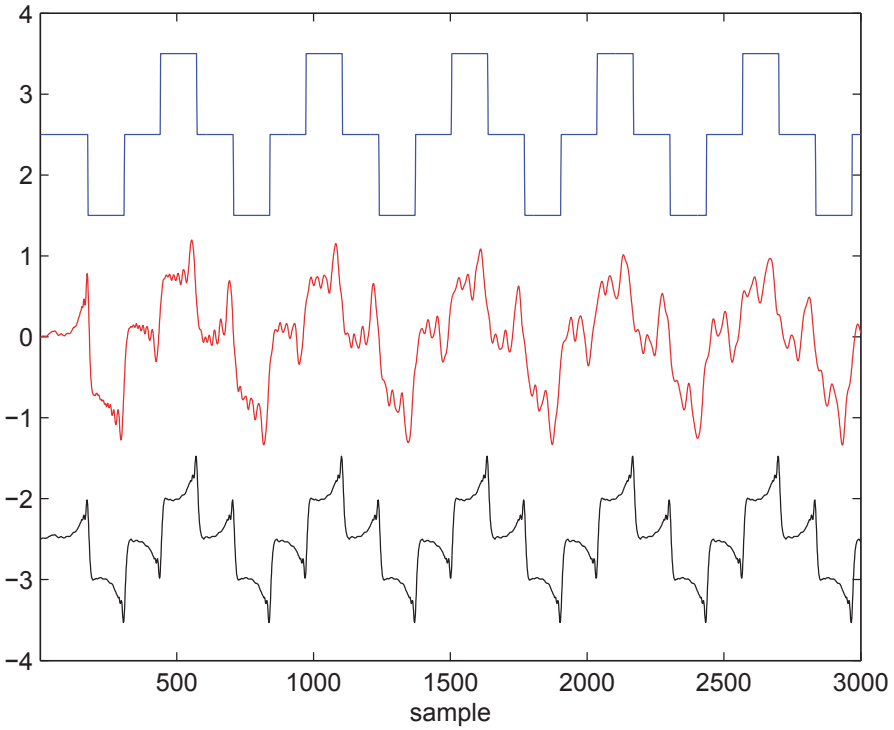

Fig. 4.11 Top: $v_{N_{pu}}(n)$. Middle: recorded signal. Bottom: synthesized signal.

can notice that the portion up to the arrow in $y'(n)$ in Fig. 4.9 corresponds to the first impulse in $a_{N_{pu}}(n)$. If we assume that the length of the pluck excitation used to generate $y(n)$ is shorter than the time interval between the first impulse and the second impulse in $a_{N_{pu}}(n)$, the indicated portion in $y'(n)$ henceforth referred to as $\tilde{a}_{exc}(n)$, then the initial plucking excitation is not distorted by reflected wave components during this time period. Therefore, we can simply extract $\tilde{a}_{exc}(n)$ by windowing it out from $y'(n)$ as shown in Fig. 4.10. One thing to note is that $\tilde{a}_{exc}(n)$ also reflects the effect of the electric pickup since we assumed the pick up transfer function $P(z) = 1$ in the DW model. Because the plucked string DW model is linear, this extracted excitation signal ($\tilde{a}_{exc}(n)$) can be used as input to the synthesis model. To validate the proposed method, we synthesized a pluck sound by convolving

$\tilde{a}_{exc}(n)$ with $v_{Npu}(n)$ and comparing it to $y(n)$. As shown in Fig. 4.11, where $v_{Npu}(n)$, $y(n)$ and the convolution of $y(n)$ and $\tilde{a}_{exc}(n)$ are depicted together, we can see that the synthesis result reproduces the signal pattern of $y(n)$ to a reasonable degree. It should be noted that this time-windowing based method would not work properly if the length of the pluck excitation in the acceleration dimension were longer than the interval between the first pulse and the second pulse in $a_{N_{pu}}(n)$, since the shape of the pluck excitation at the very beginning of the recorded signal would be distorted and smeared by the first reflection. This issue will be discussed later.

4.5 Excitation Extraction by Inverse-filtering Using the Single Delay Loop Model

4.5.1 Single Delay Loop Model Review

In [61], Karjalaninen *et al.* proposed the single delay loop (SDL) model where the conventional bidirectional DW string model is reformed to the structure having a single delay line. The SDL model can be regarded as an extension of the KS model. The SDL model can be interpreted as a source filter model in which the input is a pluck excitation of acceleration and the output can represent any variable depending on the interest. Note that the input and output of the SDL model are not traveling wave components. Prior to discussing the source-filter approach to the estimation of the finger/plectrum model parameters, we will briefly review the relation between the conventional bidirectional DW model and the SDL model. More detail can be found in [61].

The transfer function of the output velocity $v_{N_{pu}}(n)$ in Fig. 4.12 is denoted as $V_{N_{pu}}(z)$ which is given as [61],

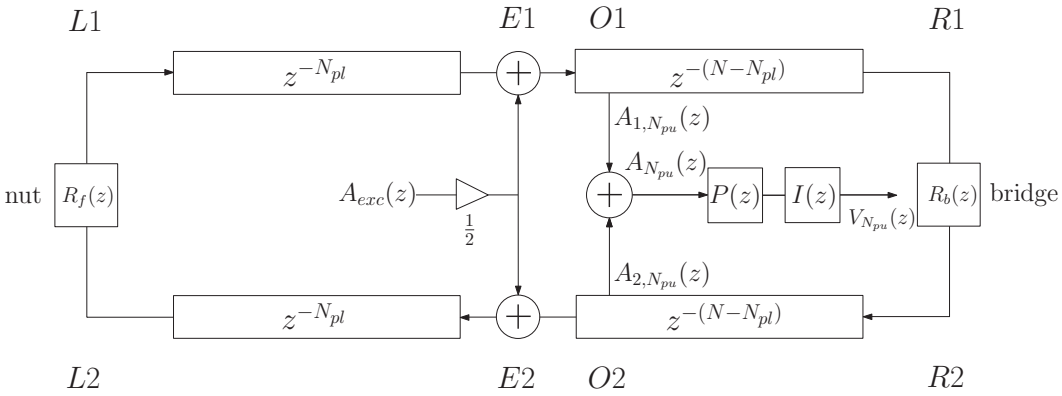

Fig. 4.12 Digital waveguide structure of the non-ideal plucked string.

$$\begin{aligned} V_{N_{pu}}(z) &= P(z)I(z)(A_{1,N_{pu}}(z) + A_{2,N_{pu}}(z)) \\ &= P(z)I(z)(A_{1,N_{pu}}(z) + H_{O1,R1}(z)R_b(z)H_{R2,O2}(z)A_{1,N_{pu}}(z)) \\ &= P(z)I(z)(1 + H_{O1,O2}(z))A_{1,N_{pu}}(z) \end{aligned} \tag{4.22}$$

$H_{O1,R1}(z)$, $H_{R2,O2}(z)$ and $H_{O1,O2}(z)$ are the transfer functions of the paths from $O1$ to $R1$, from $R2$ to $O2$ and from $O1$ to $O2$, respectively. This notational convention will be used hereafter. $A_{1,N_{pu}}(z)$ has the relation as follows [61]:

$$A_{1,N_{pu}}(z) = H_{E1,O1}(z)A_{E1,eq}(z) + H_{loop}(z)A_{1,N_{pu}}(z) \tag{4.23}$$

where,

$$H_{loop}(z) = R_b(z)H_{R2,E2}(z)H_{E2,E1}(z)H_{E1,R1}(z) \tag{4.24}$$

$$A_{E1,eq}(z) = \frac{A_{exc}(z)}{2} + H_{E2,L2}(z)R_f(z)H_{L1,E1}(z)\frac{A_{exc}(z)}{2} \tag{4.25}$$

$H_{loop}(z)$ is the transfer function representing a round trip around the loop and $A_{E1,eq}(z)$ is the transfer function representing the equivalent single excitation

at E1 [61], where $A_{exc}(z)$ is the z transform of $a_{exc}(n)$. Thus,

$$A_{1,N_{pu}}(z) = H_{E1,O1}(z)\frac{1}{1-H_{loop}(z)}A_{E1,eq}(z) \tag{4.26}$$

Substituting Eq. 4.26 into Eq. 4.22, we can obtain the overall excitation-to-

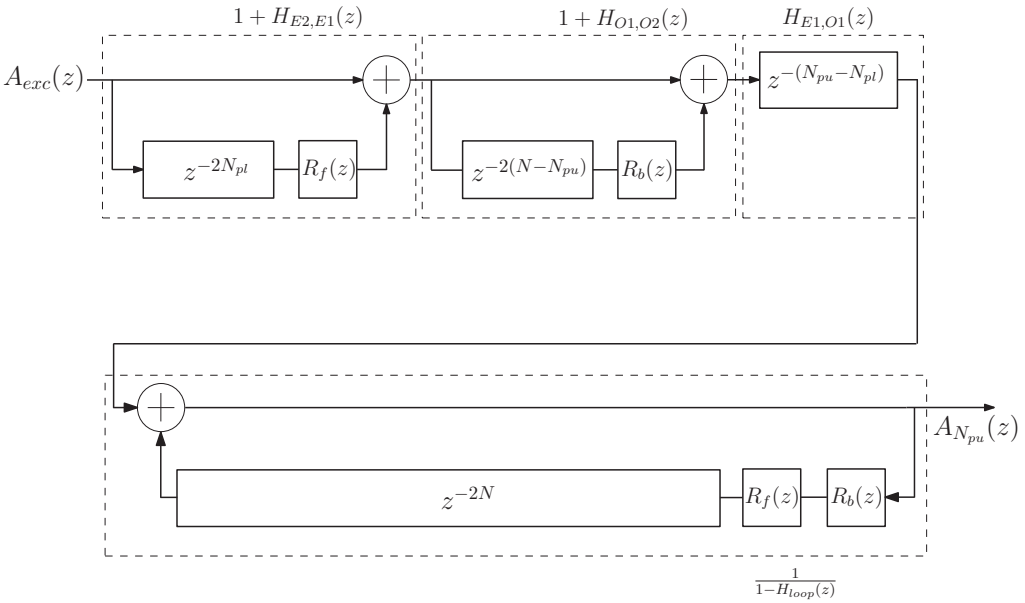

Fig. 4.13 SDL model of the plucked string.

pickup transfer function such that:

$$\begin{aligned} H(z) &= \frac{V_{N_{pu}}(z)}{A_{exc}(z)} \\ &= \frac{1}{2}[1+H_{E2,E1}(z)]\frac{H_{E1,O1}(z)}{1-H_{loop}(z)}P(z)I(z)[1+H_{O1,O2}(z)] \end{aligned} \tag{4.27}$$

As shown in Fig. 4.13, in the SDL model, the effect of the plucking position and the pickup position can be separated from the overall transfer function

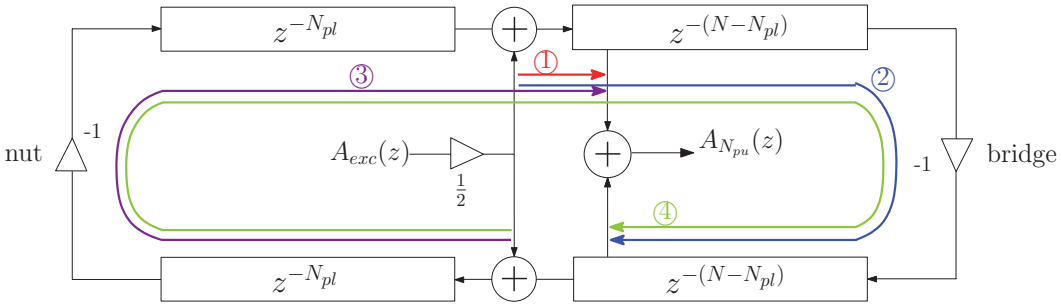

Fig. 4.14 Paths of the traveling impulses in the digital waveguide model of the ideal plucked string. The circled numbers indicate the paths of pulses in the order of arrival at the pickup position, corresponding to those in Fig.(4.15).

$H(z)$. They are characterized by the FIR transfer function,

$$H_{plpu}(z) = H_{E1,O1}(z)(1 + H_{E2,E1}(z))(1 + H_{O1,O2}(z)) \tag{4.28}$$

$$= z^{-(N_{pu}-N_{pl})}(1 + R_f(z)z^{-2N_{pl}})(1 + R_b(z)z^{-2(N-N_{pu})}) \tag{4.29}$$

$$= z^{-(N_{pu}-N_{pl})} + R_f(z)z^{-(N_{pl}+N_{pu})} \tag{4.30}$$

$$+ R_b(z)z^{-(2N-N_{pu}-N_{pl})} + R_f(z)R_b(z)z^{-(2N-N_{pu}+N_{pl})}$$

Thus Eq. 4.27 can be re-written as,

$$H(z) = \frac{H_{plpu}(z)}{2(1 - H_{loop}(z))}P(z)I(z) \tag{4.31}$$

Figure 4.15(a) illustrates the $h_{plpu}(n)$, the impulse response of $H_{plpu}(z)$ (Eq. 4.28) in the case of the ideal plucked string. The circled numbers in this figure correspond to those in Fig. 4.14 where the input excitation $a_{exc}(n)$ passes the pickup position N_{pu} either through the upper or lower delay line four times within $2N$, in the way depicted in Fig. 4.14. Note that this order is valid only when $N_{pl}+N_{pu} > N$; otherwise, the pulse on path ③ would arrive at the pickup position sooner than the one on path ② and the order should be switched. Depending on the combination of the plucking position and the pick up posi-

tion, numerous kinds of $H_{plpu}(z)$ can be created, all of which are comb filters influencing the timbre of the plucked string sound as certain modes of the vibrations are suppressed accordingly. The impulse response of $1/(1 - H_{loop}(z))$ shown in Fig. 4.15(b) is basically an impulse train of period $2N$. As can be seen in Fig. 4.15(c), the cascade of these two transfer functions characterizes the overall SDL model of the ideal plucked string and accounts for the 'bump' pattern mentioned in the previous section.

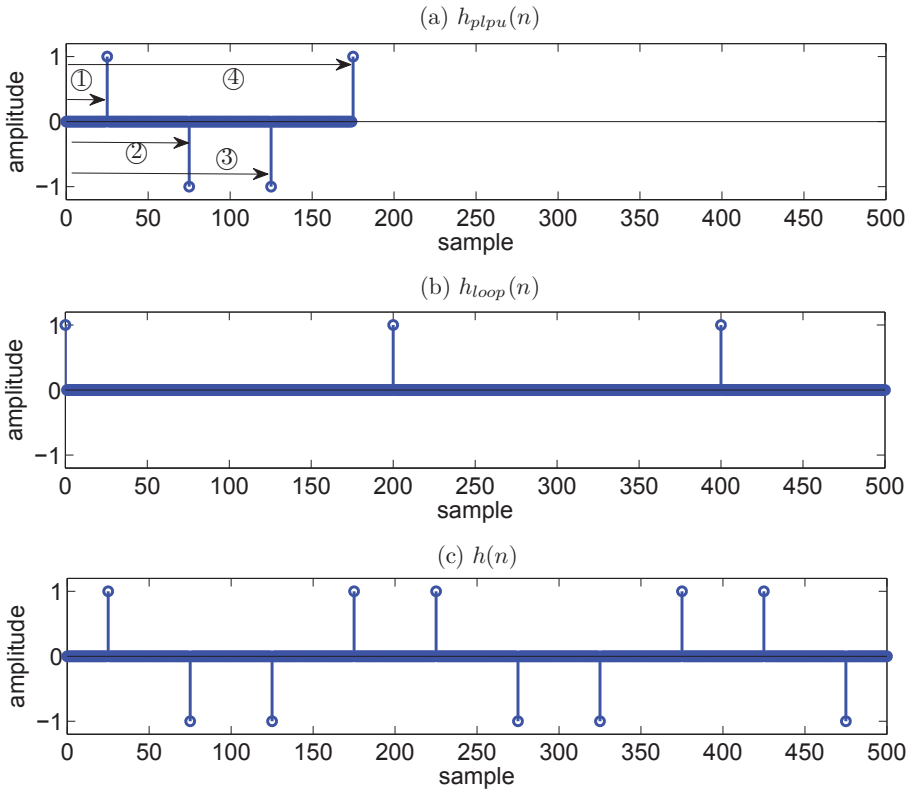

Fig. 4.15 Impulse responses of $H_{plpu}(z)$, $H_{loop}(z)$ and $H(z)$

4.6 Loop Filter Design for the DW and SDL Models

This section discusses the design of the loop filter embedded in the DW and SDL models. Contrary to the ideal plucked string model where the string vi-

bration is lossless and the traveling wave components are perfectly reflected, in real world, the string vibration undergoes damping and dispersion in a frequency-dependent manner and interacts with the bridge according to the bridge admittance. In the DW model these phenomena are typically implemented in the form of digital filters placed at the terminations that operate as reflection filters whose input and output are traveling wave variables.

4.6.1 Frequency-dependent Decay

Losses in vibrating strings are mainly caused by three different physical phenomena. Viscous drag imposed on a string, referred to as air damping, is one of them. Also, a vibrating string undergoes internal damping as a result of the material properties of the string. Lastly, a vibrating string loses its energy through the supports of the string, the bridge of an electric guitar in our case, which receives the energy from the string depending on its admittance [62]. The decaying of vibrating energy is frequency-dependent, normally in such a way that high frequency components decay faster than low frequency ones.

Much research has investigated ways of implementing frequency-dependent decay phenomena in the form of digital filters used in conjunction with DW and SDL. They are mostly based on types of measurement. They use either the measurement of frequency-dependent decay of the plucked string sound or measurement [15][2][63][64][65][66] of admittance of the supports that generally interface strings to the body of the instrument [67], or they are based on the analytic solution of the wave equation [68]. The admittance data generally accounts for the characteristics of the bridge itself and the body that interacts with the strings both acoustically and mechanically. All losses, including the reflection at the two terminations of the string, are consolidated in the loop

filter $R_b(z)$. Damping factors of harmonic partials have to be estimated. Using the short time Fourier transform (STFT), the time evolution of the amplitudes of each partial are tracked and then their decay rates are estimated. The STFT of a recorded signal $y(n)$ is

$$Y_n(k) = \sum_{m=0}^{N-1} y(hn+m)w(m)e^{-\frac{j2\pi mk}{N}}, \qquad n = 0, 1, 2, \cdots \tag{4.32}$$

where N is the DFT size, $w(m)$ is a window function, and h is the hop size. Zero padding is carried out for higher resolution in the frequency domain. The spectral peaks representing harmonic partials are detected using a peak-picking algorithm. We used the MATLAB command `findpeaks`, which is designed to find local peaks in given data. The number of harmonic partials to be considered depends on the nature of $y(n)$. By tracking the time evolution of a partial's amplitude and assuming the decay is exponential, we can use the conventional linear regression technique to fit a straight line to a time trajectory of the amplitude of the partial on a dB scale. Once the lines that fit the amplitude trajectories of partials are estimated, the slopes of those lines, β_m(dB/sample), can be derived as,

$$\beta_m = \frac{A_m(0) - A_m(n')}{n'h}, \qquad m = 1, 2, \cdots, N_{partial} \tag{4.33}$$

where $A_m(n)$ is the amplitude of the mth partial at the nth hop in dB and $N_{partial}$ is the number of partials considered. Then, the amount of the amplitude drop of each partial per string loop length, referred to as the 'loop gain',

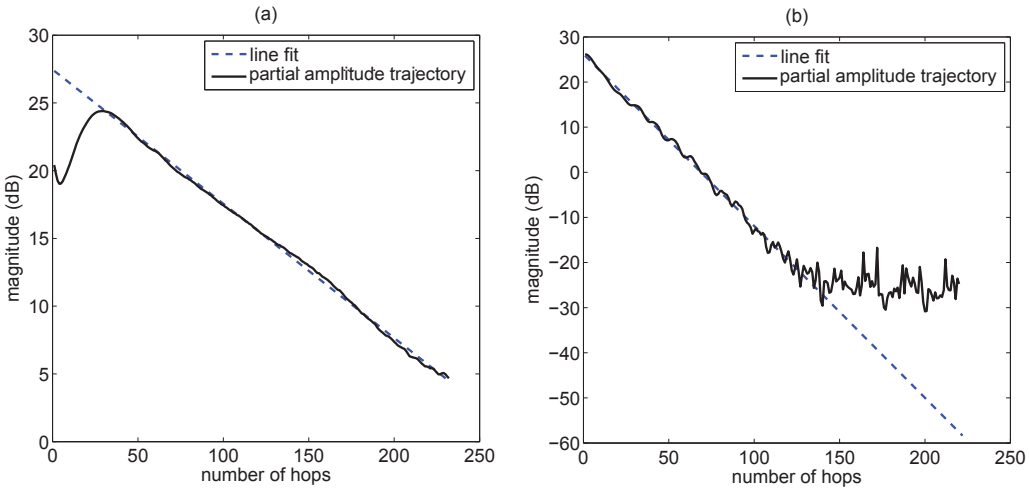

Fig. 4.16 Line fit (dashed lines) of amplitude trajectories of partials (solid lines). f_0 = 147.85 Hz and f_s=44100 Hz. The hop size is 1024 samples. (a) 2th partial (295.41 Hz). (b) 11th partial (1635 Hz).

is derived as,

$$g_m = \beta_m \frac{f_s}{f_0} = 2\beta_m N \qquad \text{(linear scale)} \tag{4.34}$$

$$G_m = 10^{g_m/20} = 10^{\beta_m N/10} \qquad \text{(dB scale)} \tag{4.35}$$

where g_m and G_m are the loop gains of the mth partial on the linear scale and on the dB scale, respectively. The phase of the loop filter target is modeled using linear phase term as:

$$P_{lin}(\omega_m) = e^{-j\omega_m} \tag{4.36}$$

where ω_m is the angular frequency of the mth partial. Thus the overall target frequency response for constructing the loop gain filter $H_{gain}(z)$ at the angular

frequencies of partials is given as

$$H_{gain,target}(\omega_m) = P_{lin}(\omega_m)G_m. \tag{4.37}$$

Measured loop gains and modeled loop phases are used to build an IIR filter in the form of

$$H_{gain}(z) = \frac{B(z)}{A(z)} = \frac{\sum_{n=0}^{N} b_n z^{-n}}{\sum_{m=0}^{M} a_m z^{-m}} \tag{4.38}$$

This is done using a MATLAB function `invfreqz` which solves the weighted least squares problem below.

$$\min_{\mathbf{b},\mathbf{a}} \sum_{m=1}^{N_{partial}} W(\omega_m)|H_{gain,target}(\omega_m)A(\omega_m) - B(\omega_m)|^2 \tag{4.39}$$

$W(\omega_m)$ is the weighting function specified at ω_m and $\mathbf{b}$, $\mathbf{a}$ are vectors of filter coefficients given as

$$\mathbf{b} = [\ b_0\ b_1\ \cdots\ b_{N-1}\ b_N\ ] \tag{4.40}$$

$$\mathbf{a} = [\ a_0\ a_1\ \cdots\ a_{M-1}\ a_M\ ] \tag{4.41}$$

Figure 4.17 depicts measured loop gains G_m at 20 partials and the magnitude response of the modeled loop gain filter $H_{gain}(z)$ based on G_m. The fundamental frequency f_0 of the targeted sound is a plucked sound of the open D string of an electric guitar with a fundamental frequency of 147.85Hz. The filter order of $H_{gain}(z)$ is $N = 1,\ M = 1$.

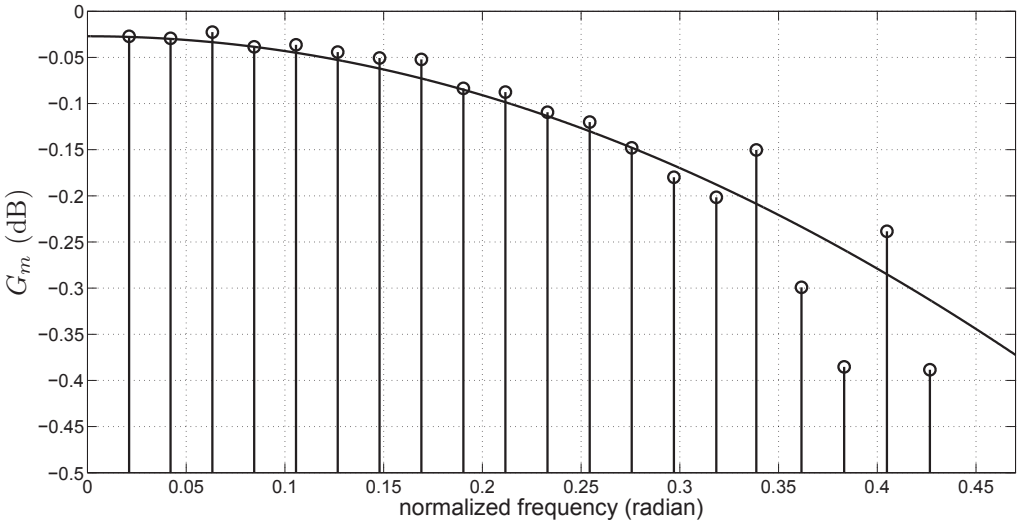

Fig. 4.17 Example of a loop gain filter. $f_0 = 147.85$ Hz and f_s=44100 Hz. The hop size is 1024 samples. Circles represent measured loop gains from partial amplitude trajectories, and a single curve represents the magnitude response of $H_{gain}(z)$, given the filter order $N = 1$, $M = 1$.

4.6.2 Dispersion

Because of the stiffness of the strings of an electric guitar, particularly the low strings, propagation of waves is frequency-dependent in the way that high frequency components travel faster than those of low frequencies [62]. Thus an impulse traveling on a stiff string becomes more and more like a 'ringing' swept sinusoid. This dispersive nature of wave propagation is an important aspect of stringed instruments that accounts for the 'inharmonicity' which often contributes to characterizing the timbre of string instruments, especially pianos. If the inharmonicity coefficient B is known, then the 'inharmonic' partial frequencies can be computed as given in [69],

$$f_k = k f_0 \sqrt{1 + Bk^2} \tag{4.42}$$

where f_0, f_k are the fundamental frequency and the frequency of the kth partial, respectively. There are many research studies proposing filter design techniques to simulate dispersion particularly in the DW model for sound synthesis [55][70][71][72][73]. They all use allpass filters to approximate frequency-dependent delays. In order to design such a dispersion filter so that it can

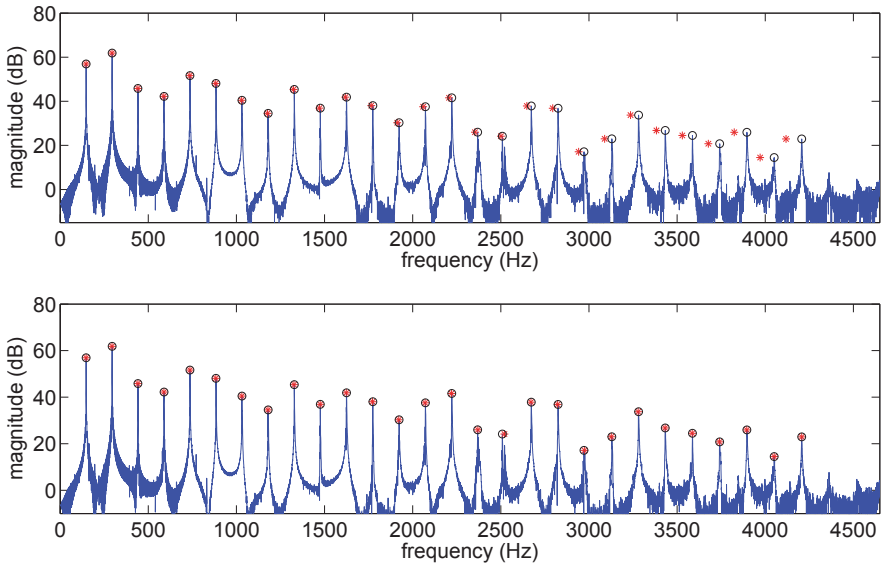

Fig. 4.18 Comparison of the spectrum of a recorded plucked string sound and the theoretical harmonics. Blue line is the magnitude response of the recorded plucked string (the low open D string of an electric guitar) sound. Black circles represent the peaks of magnitude responses. Red stars(*) are theoretical harmonics. Top: theoretical harmonics are just the multiples of the fundamental frequency. Bottom: theoretical harmonics are adjusted according to the formula of Eq. 4.42 given the estimated B.

operate in the DW model, we first estimated the inharmonicity coefficient B using the algorithm proposed in [74] where B is estimated in an iterative way given the spectrum of the target sound. Figure 4.18 shows an example of comparisons between the magnitude response of a recorded plucked string sound and the theoretically derived harmonic partials. As shown in the top pane, sounds generated by plucking a typical electric guitar string have inharmonic-

ity due to the stiffness of the string, thus the theoretical harmonics deviate from the measured harmonic spectral peaks. After applying the inharmonicity formula Eq. 4.42 given the estimated B, we can see how the theoretical harmonics better align with the measured ones.

Using the estimated B, the phase delay of the target sound is further derived. Based on this phase delay, a dispersion filter is designed using the dispersion filter design algorithm proposed in [72]. The algorithm involves designing a second-order Thiran allpass filter, and the designed allpass filter is then cascaded four times to yield the final dispersion filter as below,

$$H_{dispersion}(z) = \left(\frac{a_2 + a_1 z^{-1} + z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} \right)^4 \tag{4.43}$$

where a_1 and a_2 are the coefficients of the second-order Thiran allpass filter.

4.6.3 Loop Filter

In our model, we have lumped together the effect of $R_f(z)$ to $R_b(z)$ for the sake of convenience so that $R_f(z)$ is set as the rigid termination, simply set to -1, and $R_b(z)$ contains all the properties of lossy and stiff string vibration and the bridge. We refer to $R_b(z)$ as the loop filter hereafter. The loop filter $R_b(z)$ where all the losses and phase properties are consolidated can simply be given as below, by cascading the filters that were constructed separately:

$$R_b(z) = H_{gain}(z) H_{dispersion}(z) \tag{4.44}$$

It should be noted that the dispersion filter has a large number of delays, as its design is based on the phase delay. These delays should be compensated for by adjusting the length of the delay lines in the DW. The length of the delay lines

in the DW are adjusted with respect to the phase delay at the fundamental frequency (first harmonic) of the plucked string sound as the delay lines are supposed to determine the fundamental period.

4.7 Inverse Filtering

Once the entire SDL model has been built, we can inverse filter the recorded signal with the SDL model $H(z)$ (Eq. 4.31) as a preliminary step for extracting a pluck excitation. The inverse filtering consists of two steps. First the loop part of $H(z)$, $\frac{1}{1-H_{loop}(z)}$, is inverse filtered from the given recorded signal, and then $H_{plpu}(z)$ is inverse filtered with what remains afterwards. Prior to inverse filtering, a signal of an electric guitar captured by an electromagnetic pickup $y(n)$ is differentiated to convert $y(n)$ to an acceleration representation. The differentiation is conducted using a simple high-pass filter $D(z) = 1/I(z)$. The result of inverse filtering on $Y(z)$ with the loop part of the $H(z)$, referred to as $H_{inv}(z)$, is given as ,

$$H_{inv}(z) = \hat{A}_{exc}(z)H_{plpu}(z) = Y(z)D(z)(1 - H_{loop}(z)) \tag{4.45}$$

where $Y(z)$ is the z-transform of $y(n)$ and $\hat{A}_{exc}(z)$ is the z-transform of the pluck excitation $\hat{a}_{exc}(n)$ that we are aiming to estimate. Note that $\hat{A}_{exc}(z)$ should be distinguished from $\tilde{A}_{exc}(z)$ which represents the z-transform of the excitation $a_{exc}(n)$ obtained by the time windowing method. $H_{inv}(z)$ is the response of $H_{plpu}(z)$ when $\hat{A}_{exc}(z)$ is given as an input. As shown in Fig. 4.19, the impulse response of $H_{inv}(z)$ is the remaining signal after the effect of the loop part $1/(1 - H_{loop}(z))$ is removed from $Y(z)$.

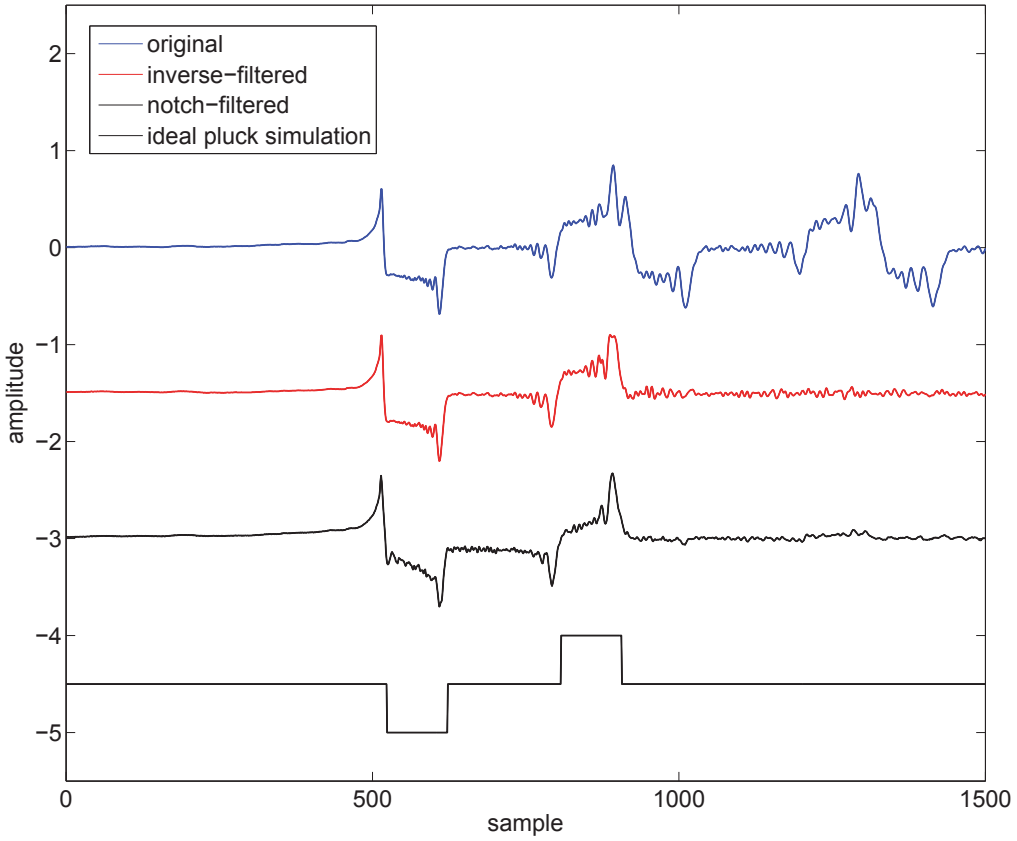

Fig. 4.19 1st: original signal. 2nd: inverse-filtered original signal. 3rd: notch-filtered original signal. 4th: the first period of the ideal plucked string SDL model.

4.7.1 Comparison to Notch Filtering

In order to verify the validity of using inverse filtering, a comparison with the results of notch filtering is discussed in this section. A notch filter to remove each harmonic peak in the spectrum of a plucked string sound is designed as

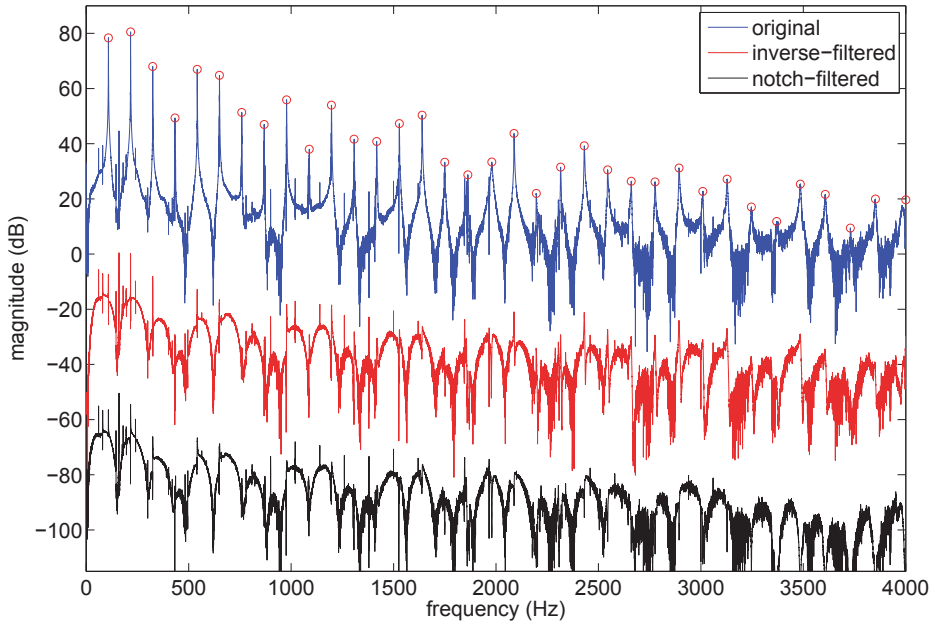

Fig. 4.20 Blue : original spectrum, Red : after notch filtering. Black : after inverse filtering using SDL model. Red circles indicated detected peaks. Spectrums are offset for comparison.

follows:

$$R = e^{-\frac{\pi BW}{fs}} \tag{4.46}$$

$$z_p = Re^{\frac{j2\pi f_p}{fs}} \tag{4.47}$$

$$H_{notch}(z) = \frac{(1 - e^{\frac{-\pi BW}{fs}} e^{\frac{j2\pi f_p}{fs}} z^{-1})(1 - e^{\frac{-\pi BW}{fs}} e^{\frac{-j2\pi f_p}{fs}} z^{-1})}{(1 - re^{\frac{-\pi BW}{fs}} e^{\frac{j2\pi f_p}{fs}} z^{-1})(1 - re^{\frac{-\pi BW}{fs}} e^{\frac{-j2\pi f_p}{fs}} z^{-1})} \tag{4.48}$$

$$H_{notch}(z) = \frac{(1 - z_p z^{-1})(1 - z_p^* z^{-1})}{(1 - r z_p z^{-1})(1 - r z_p^* z^{-1})} \tag{4.49}$$

where f_p (Hz) and BW (Hz) are the frequency of the pth peak and the associated bandwidth, respectively; z_p is the pole corresponding to the pth harmonic peak; and R is the radius of the pole z_p. The poles in the notch filter are for the isolation of notches. r is the factor that controls the amount of notch isolation. $H_{notch}(z)$ is the notch filter constructed as a second order section

based on r and z_p, targeted for removing the pth harmonic peak. The notch filter design technique given here is from [2].

Example

The signal that is notch filtered in this example was originally a sound generated by a pluck at the 21st fret on the open A string (5th string) of an electric guitar. The front pickup (the one closer to the neck) was used to capture the strings vibration. 61 notch filters were designed and applied to the given signal. Thus,

Open A, plucked at 21th fret, front pickup	
f_0	108.3 Hz
Total delay line length ($2N$)	408
pluck position N_{pl}	142
pickup position N_{pu}	154

Table 4.1 Parameters of the DW ideal plucked string model.

$$H_{E2,E1}(z) = R_f(z)z^{-284} \tag{4.50}$$

$$H_{E1,O1}(z) = z^{-12} \tag{4.51}$$

$$H_{O1,O2}(z) = R_b(z)z^{-100} \tag{4.52}$$

$$H_{loop}(z) = R_b(z)R_f(z)z^{-408} \tag{4.53}$$

$$\frac{1}{1 - H_{loop}(z)} = \frac{1}{1 - R_{rb}(z)R_f(z)z^{-408}} \tag{4.54}$$

$$\begin{aligned} H_{plpu}(z) &= [1 + R_f(z)z^{-284}]z^{-12}[1 + R_b(z)z^{-100}] \\ &= [z^{-12} + R_b(z)z^{-112} + R_f(z)z^{-296} + R_f(z)R_b(z)z^{-396}] \end{aligned} \tag{4.55}$$

$$R_f(z) = -1 \tag{4.56}$$

In Fig. 4.19, the original recorded plucked sound, the result of notch-filtering, the result of inverse filtering, and the first period of the impulse response of the SDL model of an ideal plucked (Eq. 4.54, Eq. 4.55) string are shown. Comparison of all the signals in Fig. 4.19 indicates that the similar 'bump' patterns of a single period are observed in all the signals. Fig. 4.20 shows the original magnitude response, the one after inverse-filtering, and the one after notch filtering. It appears that the result of inverse-filtering and the result of notch-filtering are in good agreement with each other. The prominent peaks are well suppressed in both residual spectra.

4.8 Extraction of Pluck Excitation Using a Recursive Least Square Algorithm

In the previous section, we introduced a method to extract a pluck excitation from a recorded plucked string sound by simply time-windowing the beginning of the recorded signal. This method allows for extracting a compact, physically meaningful pluck excitation and is also applicable to physical model-based synthesis. However, the method is valid only when a certain condition is satisfied. As the method is based on time-windowing, the shape of a desired pluck excitation should be visible in the waveform of the given signal, with its original shape preserved. In order to satisfy this condition, in the impulse response of the ideal plucked model corresponding to the given signal, the interval between the first pulse (corresponding to the path ① in Fig. 4.19) and the first reflected pulse (corresponding to the path ② in Fig. 4.19) should be longer than the length of the expected pluck excitation. Otherwise, the original shape of the expected pluck excitation could not be observed in the waveform of the given signal, as the tail of the pluck excitation directly fed

into the string would be smeared by the head of the pluck excitation which travels back after the first reflection. On the assumption that the plucking sound is the result of the convolution of the pluck excitation and the impulse response of a stringed instrument, we can 'deconvolve' the pluck excitation from the given signal even if the aforementioned condition is not satisfied. In this context, we can view how the result of the inverse filtering introduced in the previous section is the convolution of the pluck excitation and $H_{plpu}(n)$. In this section, we are proposing a method to deconvolve the pluck excitation from the result of the inverse filtering by using the RLS algorithm.

4.8.1 Recursive Least Square Algorithm

The Recursive Least Square (RLS) algorithm, or the Recursive Least Square filter, is a widely used adaptive filtering technique. Like other adaptive filters, the RLS filter takes two kinds of inputs, one of which is a desired signal $d(n)$ and the other is an input signal $u(n)$, as depicted in Fig. 4.21. The output of the RLS filter $y(n)$ can be written as

$$y(n) = \mathbf{w}^H(n)\mathbf{u}(n) \tag{4.57}$$

where $\mathbf{u}(n)$ is the input signal vector and $\mathbf{w}(n)$ is the coefficient vector of the RLS filter. The filter output is then used to obtain the difference between the desired signal $d(n)$ and $y(n)$, which is referred to as the error $e(n)$.

The cost function that the RLS algorithm attempts to minimize is the weighted least square error given in [75] as

$$\varepsilon(n) = \sum_{i=0}^{n} \lambda^{n-i}|e(i)|^2 \tag{4.58}$$

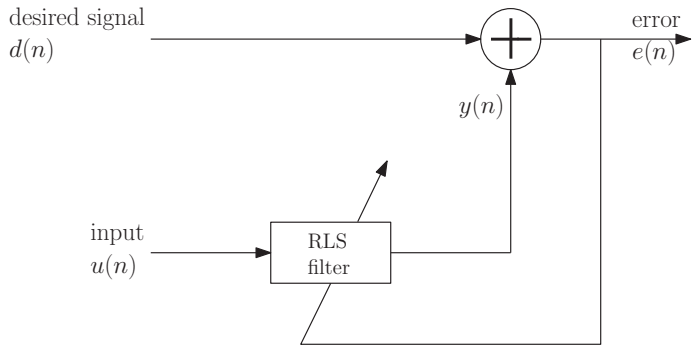

Fig. 4.21 RLS filter

where λ is the forgetting factor used to discriminately weight the error $e(i)$ in such a way that more weight is put on the more recent error. From setting $\nabla\varepsilon(n) = 0$, we obtain the optimal solution as

$$\mathbf{w}(n) = \mathbf{\Phi}(n)^{-1}\mathbf{z}(n) \tag{4.59}$$

where

$$\mathbf{\Phi}(n) = \sum_{i=0}^{n} \lambda^{n-i}\mathbf{u}(i)\mathbf{u}^{H}(i) \tag{4.60}$$

$$\mathbf{z}(n) = \sum_{i=0}^{n} \lambda^{n-i}\mathbf{u}(i)d^{*}(i) \tag{4.61}$$

are the time average correlation matrix of $\mathbf{u}(n)$ and the cross correlation vector between $\mathbf{u}(n)$ and $\mathbf{d}(n)$, respectively. Since the number of terms $e(i)$ included in the weighted least square error $\varepsilon(n)$ increases as n increases, we need to derive recursion relations for computational efficiency. Using the *matrix inversion lemma* [76] and the definitions of $\mathbf{\Phi}(n)$ and $\mathbf{z}(n)$, the recursion relations

for the RLS filter are derived as follows:

$$\begin{aligned} \mathbf{\Phi}^{-1}(n) &= \lambda^{-1}\mathbf{\Phi}^{-1}(n-1) - \lambda^{-1}\mathbf{k}(n)\mathbf{u}^H(n)\mathbf{\Phi}^{-1}(n-1) && (4.62)\\ \mathbf{z}(n) &= \lambda\mathbf{z}(n-1) + \mathbf{u}(n)d^*(n) && (4.63)\\ \mathbf{k}(n) &= \frac{\lambda^{-1}\mathbf{\Phi}^{-1}(n-1)\mathbf{u}(n)}{1+\lambda^{-1}\mathbf{u}^H(n)\mathbf{\Phi}^{-1}(n-1)\mathbf{u}(n)} && (4.64) \end{aligned}$$

where '*' denotes the hermitian operator and $\mathbf{k}(n)$ is the gain vector. Using Eqs. 4.59, 4.62, 4.63, and 4.64, we can finally derive the filter coefficients update formula in the RLS filter as

$$\mathbf{w}(n) = \mathbf{w}(n-1) + \mathbf{k}(n)[d^*(n) - \mathbf{u}^H(n)\mathbf{w}(n-1)]. \quad (4.65)$$

The main difference between the RLS algorithm and the gradient-based algorithm is that in the RLS algorithm, the input signals are used as they are, while the ensemble average of those signals is used in the gradient-based algorithm. This makes the RLS algorithm dependent on the input signal itself at every time instant, whereas the statistics of the input signals determine the behavior of the gradient-based algorithm.

4.8.2 Extraction of Pluck Using RLS Filter

The desired signal $d(n)$ used for RLS filtering is the result of inverse filtering in Eq. 4.45 as

$$D(z) = \hat{A}_{exc}(z)H_{plpu}(z) = Y(z)(1 - H_{loop}(z)) \quad (4.66)$$

where $D(z)$ is the z-transform of $d(n)$ and the output of the RLS filter $\hat{d}(n)$ is the approximation of $d(n)$ given as

$$\hat{d}(n) = \mathbf{w}^H(n)\mathbf{h}_{plpu}(n) \tag{4.67}$$

where $\mathbf{w}(n)$, $\mathbf{h}_{plpu}(n)$ are vectors of size $(N_{exc} \times 1)$ defined as,

$$\mathbf{w}(n) = [w(1,n) \quad w(2,n) \quad \cdots \quad w(N-1,n) \quad w(N,n)]^T \tag{4.68}$$

$$\mathbf{h}_{plpu}(n) = [h_{plpu}(n) \quad h_{plpu}(n-1) \quad \cdots \quad h_{plpu}(n-(N_{exc}-2)) \quad h_{plpu}(n-(N_{exc}-1))]^T \tag{4.69}$$

and N_{exc} is the length of $\mathbf{w}(n)$. Note that in the RLS filtering in Eq. 4.67, $h_{plpu}(n)$ corresponds to the input signal and $\mathbf{w}(n)$ corresponds to the coefficient vector of the RLS filter. $w(k,n)$ denotes the kth coefficient of $\mathbf{w}(n)$ at the nth iteration. The optimal solution for Eq. 4.67 is

$$\mathbf{w}(n) = \mathbf{\Phi}(n)^{-1}\mathbf{z}(n) \tag{4.70}$$

where $\mathbf{\Phi}(n)$ and $\mathbf{z}(n)$ are matrices of sizes $(N_{exc} \times N_{exc})$ and $(N_{exc} \times 1)$, respectively, defined as

$$\mathbf{\Phi}(n) = \sum_{i=0}^{N_{exc}} \lambda^{n-i}\mathbf{h}_{plpu}(i)\mathbf{h}_{plpu}^H(i) \tag{4.71}$$

$$\mathbf{z}(n) = \sum_{i=0}^{N_{exc}} \lambda^{n-i}\mathbf{h}_{plpu}(i)d^*(i) \tag{4.72}$$

By updating the recursion relations below in the same way as in Eq. 4.62 $\sim$ Eq. 4.64,

$$\mathbf{\Phi}^{-1}(n) = \lambda^{-1}\mathbf{\Phi}^{-1}(n-1) - \lambda^{-1}\mathbf{k}(n)\mathbf{h}_{plpu}^{H}(n)\mathbf{\Phi}^{-1}(n-1) \tag{4.73}$$

$$\mathbf{z}(n) = \lambda\mathbf{z}(n-1) + \mathbf{h}_{plpu}(n)d^{*}(n) \tag{4.74}$$

$$\mathbf{k}(n) = \frac{\lambda^{-1}\mathbf{\Phi}^{-1}(n-1)\mathbf{h}_{plpu}(n)}{1+\lambda^{-1}\mathbf{h}_{plpu}^{H}(n)\mathbf{\Phi}^{-1}(n-1)\mathbf{h}_{plpu}(n)} \tag{4.75}$$

we can derive the RLS filter update equation for $\mathbf{w}(n)$ as

$$\mathbf{w}(n) = \mathbf{w}(n-1) + \mathbf{k}(n)[d^{*}(n) - \mathbf{h}_{plpu}^{H}(n)\mathbf{w}(n-1)] \tag{4.76}$$

By using the RLS algorithm, one can observe how the estimate of pluck excitation $\mathbf{w}(n)$ varies as updated.

To validate the algorithm, an ideal plucked string sound is synthesized by feeding a synthesized excitation into a DW model. For the synthesized excitation, a Hann window of 50 samples long is used. Figure 4.22 illustrates $h_{plpu}(n)$ of the ideal DW model and the synthesized one. Figure 4.23 illustrates the time updates of the RLS filter. As updating goes on, the updated RLS filter becomes more like the first bump, a Hann window, in $h_{plpu}(n)$ of the synthesized signal, here from the sixth pattern. It shows that, for an ideal case, the RLS is able to extract the excitation perfectly. For another example, an extraction using the RLS algorithm is applied to a recorded plucked-string sound. As the first 'bump' in $d(n)$ (Fig. 4.24) is the least distorted shape of the pluck excitation, the estimated $\mathbf{w}(n)$ at the early stage of updates may possibly be the desired estimate. Note that the first and the third bumps in $h_{plpu}(n)$ are ideal impulses, as depicted in Fig. 4.24. This is because the first bump did not go through any flipping from one delay line to the other delay

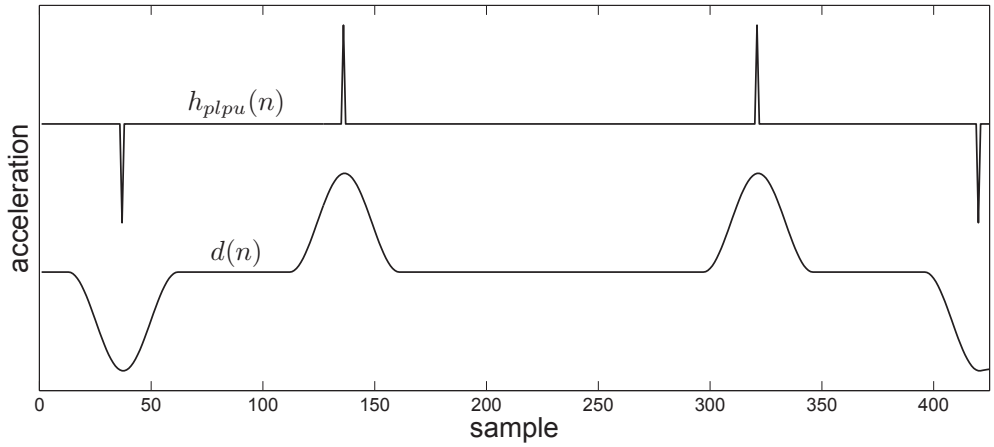

Fig. 4.22 $h_{plpu}(n)$ and $d(n)$ of the synthesized plucked string sound using an ideal DWG model and a Hann window.

line at either end, which corresponds to path ① in Fig. 4.14 and Fig. 4.15; and the third bump corresponding to path ③ in Fig. 4.14 and Fig. 4.15, is flipped at the 'nut' side where only the phase inversion occurs as all the losses and the dispersion are consolidated into the filter $R_b(z)$ located at the other end, as defined in the previous section.

Figure 4.25 depicts the time updates of the RLS filter. Different from the result for an ideal plucked string case, from the seventh pattern, the shape of the RLS filter coefficients start becoming different from the first bump pattern in the recorded signal. This is because $h_{plpu}(n)$ does not perfectly describe the true waveform of the first cycle of the recorded signal. Figure 4.26 and 4.27 show examples of pluck excitations extracted by using the method introduced in this section. The first three examples from the top in Fig. 4.26 have a common feature in that all of them first gradually rise, then undergo sharp falls, and then return to zero. Also, the magnitudes gradually decrease after short rises in the low frequency region. These three examples are extracted from sound samples that were created by down-picking a string with a plectrum generally used for playing an electric guitar. The angle of plucking is some-

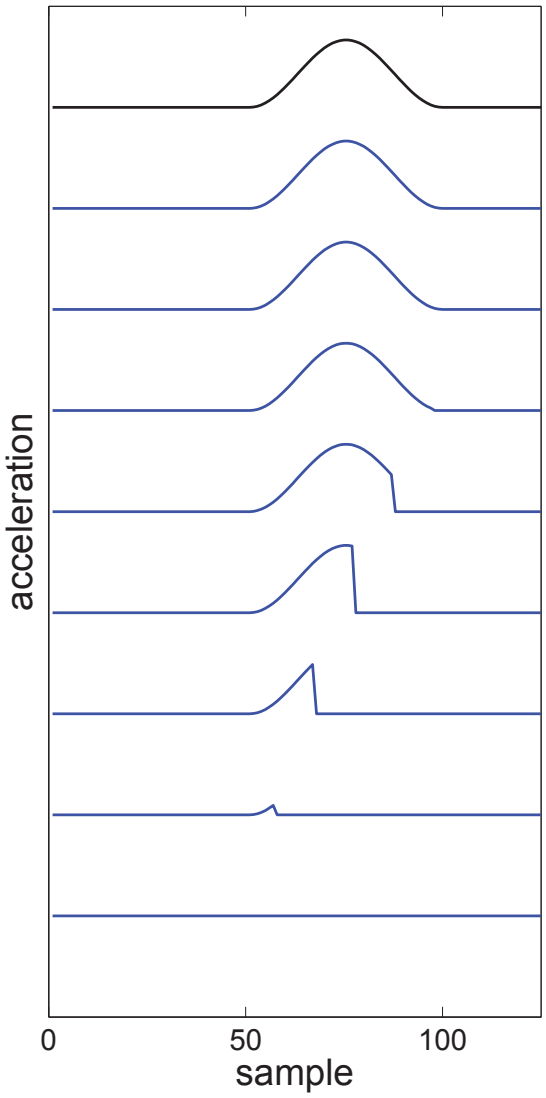

Fig. 4.23 Updates of the RLS filter $\mathbf{w}(n)$, temporally updated from the bottom to the top. The black one at the very top is the original signal.

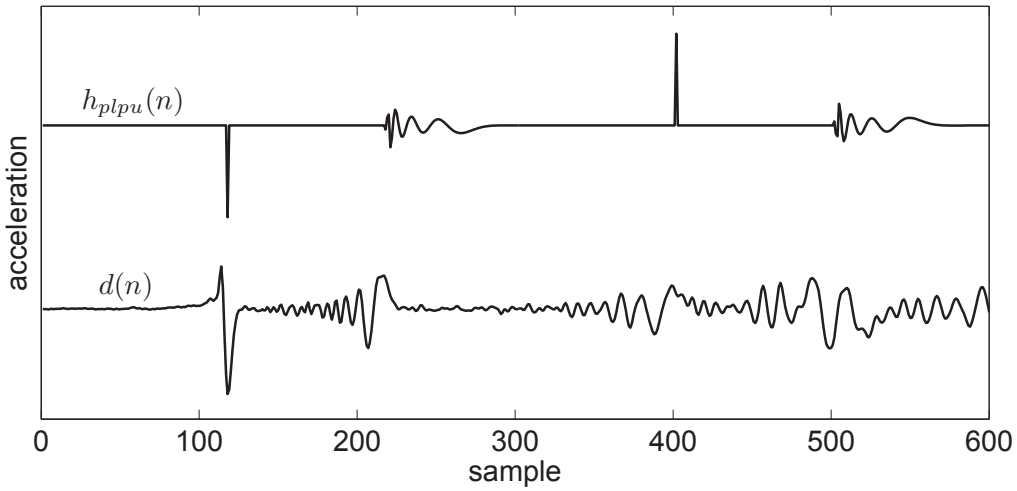

Fig. 4.24 $h_{plpu}(n)$ and $d(n)$. Amplitudes (acceleration) are normalized for the comparison.

where between one perpendicular to the body of the guitar and one parallel to the body in the direction away from the lowest string toward the highest string. The last example in Fig. 4.26 and the second example in Fig. 4.27 depict pluck excitations obtained from sounds where a player especially tried to pluck as perpendicularly as possible to the guitar's body with up-picking. In these examples, we can see that the extracted pluck excitations do not show the phase changes (the gradual rises followed by sharp downfalls, then return to zero) that appeared in the excitations previously mentioned, but rather show gradual rises and falls. Based on these observations, it can be conjectured that the electromagnetic pickup senses a fluctuation differently in relation to the angle of fluctuation. The first example in Fig. 4.27 is from a pluck sound created by plucking with the thumb using the same plucking angle as the first three excitations we discussed. This sound is the 'dullest' since the thumb is in contact with the string longer than the duration a plectrum is in contact with the string; thus, the width of the impulsive portion in the excitation signal is longer than others and, in the magnitude response, the magnitude decreases

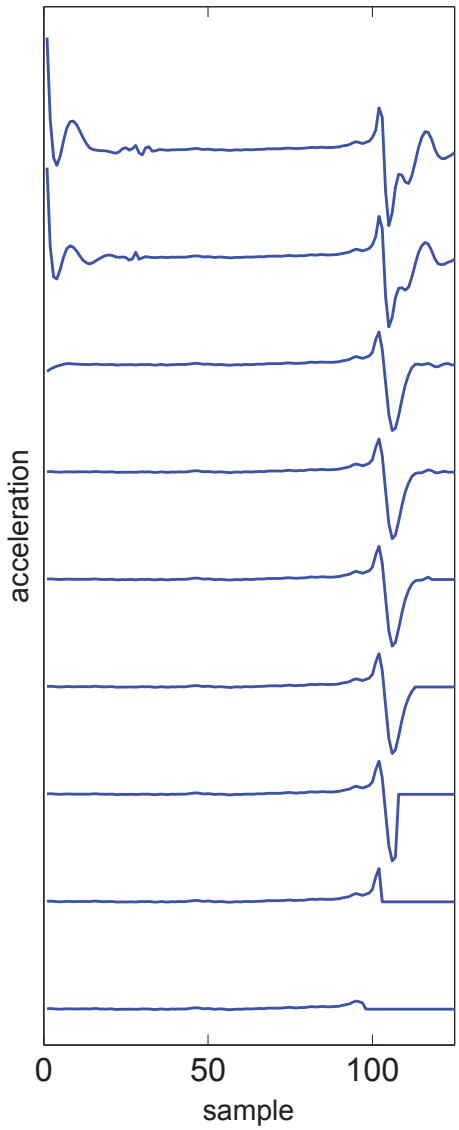

Fig. 4.25 Updates of the RLS filter $\mathbf{w}(n)$, temporally updated from the bottom to the top.

most quickly as the frequency increases, compared to other excitations. The last example in Fig. 4.27 is from the sound generated by plucking upwardly a string with a finger at the angle that is perpendicular to the guitar body in the same way as in the last example in Fig. 4.26 and the second example in Fig. 4.27. An interesting point to note is the way the excitation shape evolves. It shows an abrupt sharp peak neighbored by valleys. This may be explained by the way the finger interacts with the string while plucking. A string is in contact with a finger facing upward before the finger moves to begin a pluck action; and as the pluck action begins, the string slips away from the finger with acceleration; just before the string is completely released from the finger's flesh, it is hit by the fingernail. This may explain why we see an abrupt sharp uprise in the excitation shape. Sound examples synthesized using extracted excitations are available on-line[2]. Various combinations of extracted plucking excitations and the SDL models with different parameters (fundamental frequencies, pickup and plucking positions) can also be found.

4.9 Parametric Model of Pluck and Estimation

In this section, we will discuss parametric modeling of an extracted pluck excitation and estimation of parameters given the extracted pluck excitation. Considering typical shapes of extracted excitations, an attempt to use a parametric model originally developed to describe a glottal flow derivative is created for these tasks. Among many models, the well-known Liljencrants-Fant (LF) model [77] is chosen to fit the pluck excitations, and parameter estimation is conducted using the extended Kalman filter [78] since the LF model is a non-linear model.

[2]http://www.music.mcgill.ca/∼lee/pluckexcitation

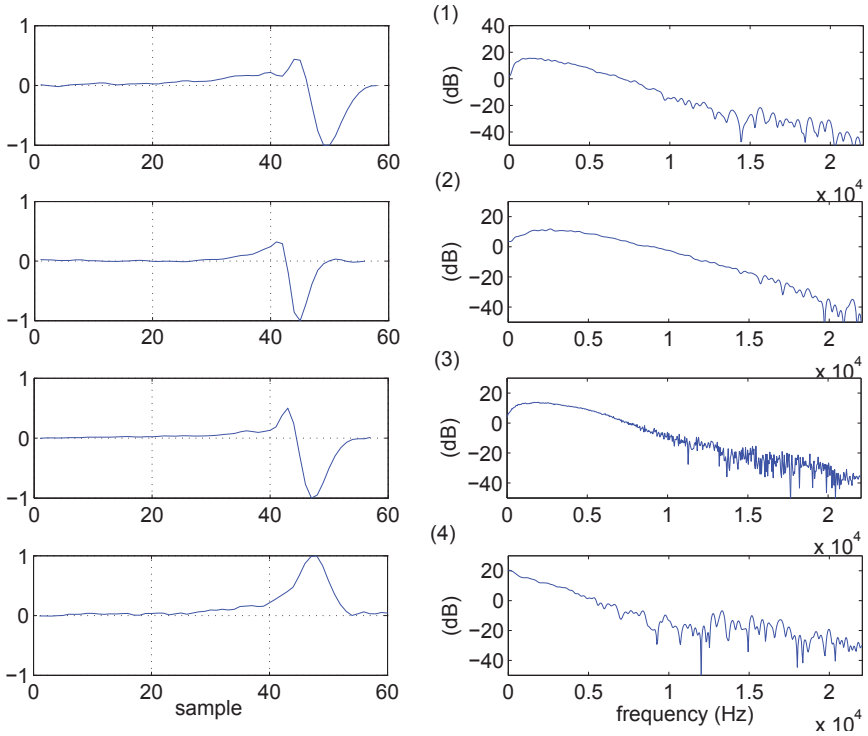

Fig. 4.26 Examples of extracted excitations. Figures on the left side are extracted excitations and those on the right side are the spectra of the extracted excitations. Amplitudes of extracted excitations are normalized for the comparison.

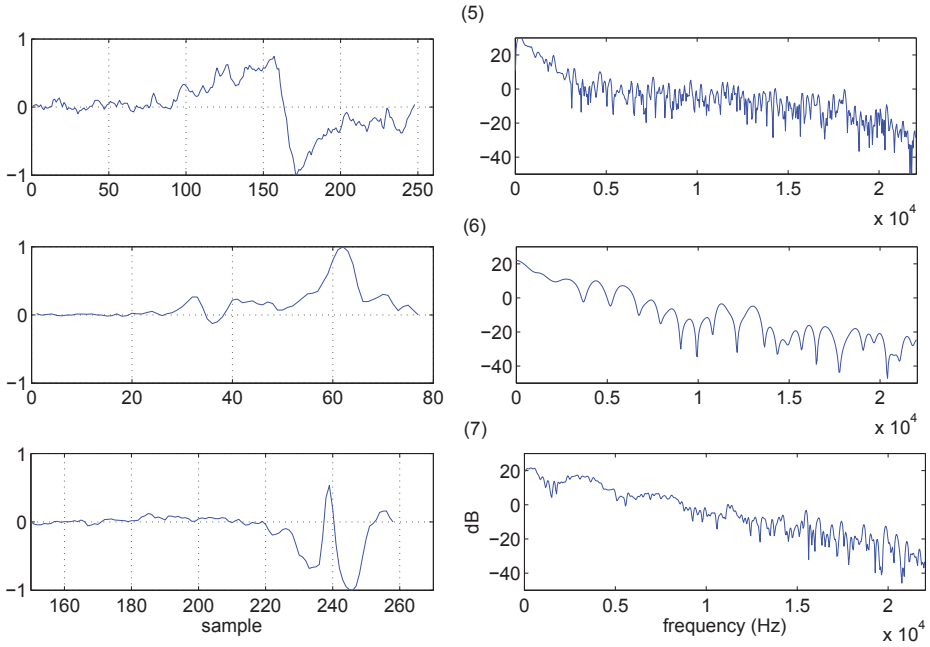

Fig. 4.27 Examples of extracted excitations (cont'd.)

4.9.1 Liljencrants-Fant Model

The Liljencrants-Fant (LF) model is one of the most widely used parametric models. It describes one period of the glottal flow derivative (GFD) waveform. The LF model involves four parameters that determine the waveform of the GFD. Those four parameters are time indexes t_e, t_p, t_a and the amplitude E_e. If one cycle of the GFD is t_c, then the GFD $g(t)$ is determined by the LF model as

$$g(t) = \begin{cases} E_0 e^{\alpha t} sin(\omega_g t), & 0 \leq t \leq t_e \\ -\dfrac{E_e}{\epsilon t_a}[e^{-\epsilon(t-t_e)} - e^{-\epsilon(t_c-t_e)}], & t_e < t \leq t_c. \end{cases} \tag{4.77}$$

E_0, ω_g, α and ϵ are derived as

$$\omega_g = \frac{\pi}{t_p} \tag{4.78}$$

$$\epsilon t_a = 1 - e^{-\epsilon(t_c-t_e)} \tag{4.79}$$

$$\alpha = \frac{t_c - t_e}{e^{\epsilon(t_c-t_e)} - 1} - \frac{1}{\epsilon} \tag{4.80}$$

$$E_0 = -\frac{E_e}{e^{\alpha t_e} sin(\omega_g t_e)} \tag{4.81}$$

As shown in Fig. 4.28, E_e is the amplitude of the minimum of the GFD at t_e, the glottal closing instant, and t_a is the time constant representing the return phase in terms of how quickly the GFD comes back to zero from its minimum point. t_p is the instant where the glottal flow reaches its maximum. t_0 is the starting point of the cycle and is set at 0. Parameters E_0, ω_g, and α describe the shape of the open phase, and E_e and ϵ describe the shape of the return phase.

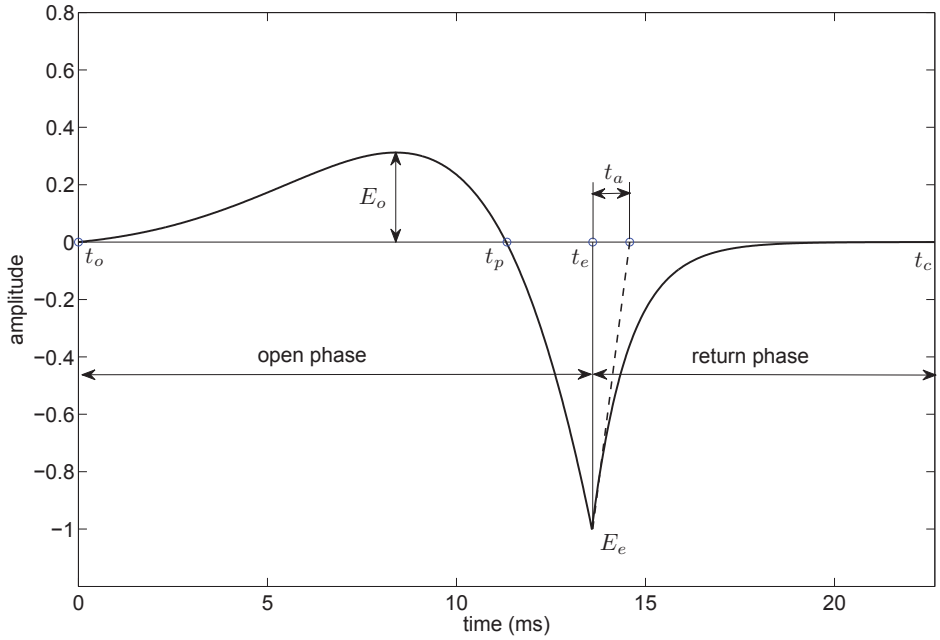

Fig. 4.28 LF model

4.9.2 Parameter Estimation of LF model Using the Extended Kalman Filter

The Extended Kalman filter

The extended Kalman filter (EKF) is a variant of the Kalman filter that can be used when a given state space model is non-linear. Basically, the EKF approximates the non linearity in the given non-linear state space model by 'linearizing' it, resulting in a linear state space model; and it then applies the original Kalman filtering in order to estimate the state from observed samples. The non-linear state space model is described as [75]

$$\mathbf{x}(n+1) = \mathbf{F}(n, \mathbf{x}(n))\mathbf{x}(n) + \mathbf{u}(n) \tag{4.82}$$

$$\mathbf{y}(n) = \mathbf{C}(n, \mathbf{x}(n))\mathbf{x}(n) + \mathbf{v}(n) \tag{4.83}$$

Equation 4.82 is the state equation in which $\mathbf{x}(n)$ is the state vector whose dimension is assumed to be $(M \times 1)$; and $\mathbf{F}(n, \mathbf{x}(n))$ is the transition matrix of dimension $(M \times M)$, which is non-linear. The state equation is driven by a zero-mean, white noise vector $\mathbf{u}(n)$ of size $(M \times 1)$. Equation 4.83 is the observation equation. $\mathbf{y}(n)$ is the observation vector assumed to be of dimension $(N \times 1)$, which is corrupted by the $(N \times 1)$ white noise vector $\mathbf{v}(n)$; and the $(N \times M)$ vector $\mathbf{C}(n, \mathbf{x}(n))$ is the measurement matrix, which is also non-linear. White noise vectors $\mathbf{u}(n)$ and $\mathbf{v}(n)$ are assumed to be independent of each other so that

$$E[\mathbf{u}(n)\mathbf{u}^H(m)] = \begin{cases} \mathbf{Q}(n), & n = m \\ \mathbf{0}, & n \neq m \end{cases} \tag{4.84}$$

$$E[\mathbf{v}(n)\mathbf{v}^H(m)] = \begin{cases} \mathbf{R}(n), & n = m \\ \mathbf{0}, & n \neq m \end{cases} \tag{4.85}$$

where $\mathbf{Q}(n)$, $\mathbf{R}(n)$ are the correlation matrices of $\mathbf{u}(n)$, $\mathbf{v}(n)$, respectively. Also, it is assumed that $\mathbf{u}$(n) is independent of $\mathbf{y}(n)$, and $\mathbf{v}(n)$ is independent of both $\mathbf{x}(n)$ and $\mathbf{y}(n)$.

The EKF recursively finds the estimate of the state $\mathbf{x}(n)$ in the minimum mean square error (**MMSE**) sense. The *a priori* estimate (prediction) of the state at time $n+1$ given the observation up to n is,

$$\hat{\mathbf{x}}(n+1|n) = \mathbf{F}(n, \hat{\mathbf{x}}(n|n)) \tag{4.86}$$

where $\hat{\mathbf{x}}(n|n)$ is the *a posteriori* estimate (filtering) of the state at time n given

the observation up to n, which is recursively obtained as

$$\hat{\mathbf{x}}(n|n) = \hat{\mathbf{x}}(n|n-1) + \mathbf{G}_f(n)[\mathbf{y}(n) - \mathbf{C}(n, \hat{\mathbf{x}}(n|n-1))]. \tag{4.87}$$

$\mathbf{G}_f(n)$ is the gain matrix for the EKF, which can also be recursively derived as

$$\mathbf{G}_f(n) = \mathbf{K}(n, n-1)\mathbf{C}^H(n)[\mathbf{C}(n) + \mathbf{K}(n, n-1)\mathbf{C}^H(n) + \mathbf{R}(n)]^{-1} \tag{4.88}$$

where $\mathbf{C}(n)$ is the newly defined matrix that is a partial derivative of $\mathbf{C}(n, \mathbf{x}(n))$ with respect to $\mathbf{x}(n)$, evaluated at $\hat{\mathbf{x}}(n|n-1)$ as

$$\mathbf{C}(n) = \frac{\partial \mathbf{C}(n, \mathbf{x})}{\partial \mathbf{x}}\Big|_{\mathbf{x}=\hat{\mathbf{x}}(n|n-1)}. \tag{4.89}$$

and $\mathbf{K}(n, n-1)$ is referred to as the predicted state-error correlation matrix given as,

$$\mathbf{K}(n, n-1) = E[(\mathbf{x}(n) - \hat{\mathbf{x}}(n|n-1))(\mathbf{x}(n) - \hat{\mathbf{x}}(n|n-1))^H]. \tag{4.90}$$

$\mathbf{K}(n, n-1)$ is updated as,

$$\mathbf{K}(n, n-1) = \mathbf{F}(n, n-1)\mathbf{K}(n-1)\mathbf{F}^H(n, n-1) + \mathbf{Q}(n-1) \tag{4.91}$$

$\mathbf{F}(n, n-1)$ is also newly defined for the EKF in the same way as $\mathbf{C}(n)$ is in Eq. 4.89, but it is evaluated at $\hat{\mathbf{x}}(n-1|n-1)$:

$$\mathbf{F}(n, n-1) = \frac{\partial \mathbf{F}(n, \mathbf{x})}{\partial \mathbf{x}}\Big|_{\mathbf{x}=\hat{\mathbf{x}}(n-1|n-1)}. \tag{4.92}$$

$\mathbf{K}(n)$ is the filtered state-error correlation matrix defined as,

$$\mathbf{K}(n) = E[(\mathbf{x}(n) - \hat{\mathbf{x}}(n|n))(\mathbf{x}(n) - \hat{\mathbf{x}}(n|n))^H] \tag{4.93}$$

and also recursively estimated as,

$$\mathbf{K}(n) = [I - \mathbf{G}_f(n)\mathbf{C}(n)]\mathbf{K}(n, n-1) \tag{4.94}$$

With the initial conditions given below,

$$\hat{\mathbf{x}}(1|0) = E[\mathbf{x}(1)] \tag{4.95}$$

$$\mathbf{K}(1,0) = E[(\mathbf{x}(1) - E[\mathbf{x}(1)])(\mathbf{x}(1) - E[\mathbf{x}(1)])^H] \tag{4.96}$$

and using Eq. 4.84 $\sim$ Eq. 4.94 which recursively update variables as a new observation is fed in, the optimal estimate of state vector $\mathbf{x}(n)$ can be obtained.

Estimation of LF parameters using the EKF

This section deals with the estimation of LF parameters in the case of an extracted pluck excitation. To fit the LF model to an extracted pluck excitation obtained from the RLS algorithm, $\mathbf{w}(n)$, the LF parameter estimation method proposed in [78] is used. This method employs the EKF to recursively estimate the LF parameters. The discrete time signal version of the LF model is given in [78] , with $t_c = 1$, as

$$g(n) = \begin{cases} -\dfrac{E_e}{sin(\dfrac{\pi T_e}{T_p})} e^{-\alpha(T_e - \frac{n}{N})} sin(\dfrac{\pi n}{T_p N}), & 0 \le n \le T_e N \\ -\dfrac{E_e}{\epsilon T_a}[e^{-\epsilon(\frac{n}{N} - T_e)} - e^{-\epsilon(1-T_e)}], & T_e N \le n \le N \end{cases} \tag{4.97}$$

where $T_e = t_e/t_c$, $T_p = t_p/t_c$ and $T_a = t_a/t_c$, respectively. For an extracted pluck excitation $w(n)$, the estimated RLS filter coefficient, two separate EKFs are employed, one for the open phase ($0 \leq n \leq T_eN$) and the other for the return phase ($T_eN \leq n \leq N$) (Fig. 4.28). In order to estimate α, the state space model that describes the waveform of GFD during the open phase is constructed as

$$\alpha(n) = \alpha(n-1) \tag{4.98}$$

$$g(n) = \mathbf{C}_o(n, \alpha(n)) + q(n) \tag{4.99}$$

where $q(n)$ is a white noise process and the transition matrix $\mathbf{C}_o(n, \alpha(n))$ of size (1×1) is given from Eq. 4.97 as,

$$\mathbf{C}_o(n, \alpha(n)) = -\frac{E_e}{sin(\frac{\pi T_e}{T_p})} e^{-\alpha(T_e - \frac{n}{N})} sin(\frac{\pi n}{T_p N}) \tag{4.100}$$

With the time derivative of $\mathbf{C}_o(n, \alpha(n))$ evaluated at $\hat{\alpha}(n|n-1)$ given as,

$$\mathbf{C}_o(n) = \frac{\partial \mathbf{C}_o(n, \alpha(n))}{\partial \alpha}\Big|_{\alpha=\hat{\alpha}(n|n-1)} \tag{4.101}$$

$$= \frac{E_e}{sin(\frac{\pi T_e}{T_p})} (T_e - \frac{n}{N}) e^{-\hat{\alpha}(n|n-1)(T_e - \frac{n}{N})} sin(\frac{\pi n}{T_p N}) \tag{4.102}$$

and assuming that T_e and T_p are known, we can evaluate $\mathbf{C}_0(n)$ and $\mathbf{C}_0(n, \alpha(n))$. t_p can generally be found at the first zero-crossing point ahead of t_e. Accord-

ingly, the EKF update equations for the open phase can be derived as follows:

$$\mathbf{G}_f(n) = \mathbf{K}(n, n-1)\mathbf{C}_o^H(n)[\mathbf{C}_o(n) + \mathbf{K}(n, n-1)\mathbf{C}_o^H(n) + \mathbf{Q}(n)]^{-1} \quad (4.103)$$

$$\hat{\alpha}(n|n) = \hat{\alpha}(n|n-1) + \mathbf{G}_f(n)[w(n) - \mathbf{C}_o(n, w(n|n-1))] \quad (4.104)$$

$$\mathbf{K}(n) = [I - \mathbf{G}_f(n)\mathbf{C}_o(n)]\mathbf{K}(n, n-1). \quad (4.105)$$

For initial conditions, we set $\mathbf{K}(1, 0) = 1$ following the results of internal experiments and the suggestion from [78]. $\mathbf{Q}(n)$ is estimated from the portion of the extracted pluck excitation where noise is dominant and relatively little meaningful information is present. This is usually a portion ahead of the beginning of the pluck excitation. The initial value for $\hat{\alpha}$, $\alpha(1|0)$, is determined within the range 1-100 as proposed in [78]. To find the optimum $\alpha(1|0)$, the EKF is run using an $\alpha(1|0)$. An estimated $\hat{\alpha}$ is obtained and then the minimum square error (**MSE**) between the $w(n)$ and the synthesized excitation is calculated using the obtained $\hat{\alpha}$. This process is repeated for all the integer numbers within the range (1-100) for $\alpha(1|0)$. Among all the tried $\alpha(0|1)$, we pick the $\alpha(1|0)$ that yields the smallest **MSE** between $w(n)$ and the synthesized excitation. Once all initial conditions are set, we can recursively estimate ϵ using the EKF updates and the state space relation in Eqs. 4.98 and 4.99 as below,

$$\hat{\alpha}(n|n-1) = \hat{\alpha}(n-1|n-1) \quad (4.106)$$

$$\mathbf{K}(n, n-1) = \mathbf{K}(n-1). \quad (4.107)$$

In the same way as the open phase, the state space model describing the waveform of GFD during the return phase is constructed as

$$\epsilon(n) = \epsilon(n-1) \tag{4.108}$$

$$g(n) = \mathbf{C}_r(n, \epsilon(n)) + v(n), \tag{4.109}$$

where $v(n)$ is a white noise process and the transition matrix $\mathbf{C}_r(n, \epsilon(n))$ is given from Eq. 4.97 as

$$\mathbf{C}_r(n, \epsilon(n)) = -\frac{E_e}{\epsilon T_a}[e^{-\epsilon(\frac{n}{N}-T_e)} - e^{-\epsilon(1-T_e)}]. \tag{4.110}$$

With the time derivative of $\mathbf{C}_r(n, \epsilon(n))$ evaluated at $\hat{\epsilon}(n|n-1)$ given as

$$\mathbf{C}_r(n) = \frac{\partial \mathbf{C}_r(n, \epsilon(n))}{\partial \epsilon}\Big|_{\epsilon=\hat{\epsilon}(n|n-1)} \tag{4.111}$$

$$= \frac{E_e}{\hat{\epsilon}(n|n-1)T_a}[(\frac{1}{\hat{\epsilon}(n|n-1)} - \frac{n}{N} - T_e)e^{-\hat{\epsilon}(n|n-1)(\frac{n}{N}-T_e)} \tag{4.112}$$

$$- (\frac{1}{\hat{\epsilon}(n|n-1)} + 1 - T_e)e^{-\hat{\epsilon}(n|n-1)(1-T_e)}].$$

the EKF update equations for the return phase can be derived as follows:

$$\mathbf{G}_f(n) = \mathbf{K}(n, n-1)\mathbf{C}_r^H(n)[\mathbf{C}_r(n) + \mathbf{K}(n, n-1)\mathbf{C}_r^H(n) + \mathbf{V}(n)]^{-1} \tag{4.113}$$

$$\hat{\epsilon}(n|n) = \hat{\epsilon}(n|n-1) + \mathbf{G}_f(n)[\hat{a}_{exc}(n) - \mathbf{C}_r(n, \hat{\epsilon}(n|n-1))] \tag{4.114}$$

$$\mathbf{K}(n) = [I - \mathbf{G}_f(n)\mathbf{C}_r(n)]\mathbf{K}(n, n-1). \tag{4.115}$$

Just as with the open phase, we initially have $\mathbf{K}(1, 0) = 1$, and $\mathbf{V}(n)$ is estimated in the same way as $\mathbf{Q}(n)$ is estimated. $\epsilon(1|0)$ is also determined in the same way that $\alpha(1|0)$ is determined but here the range for $\epsilon(1|0)$ is 1-200, as suggested in [78]. Once all initial conditions are set, we can recursively esti-

mate ϵ using the EKF updates (Eq. 4.113 $\sim$ Eq. 4.115) and the state space relation in Eqs. 4.108 and 4.109 as follows:

$$\hat{\epsilon}(n|n-1) = \hat{\epsilon}(n-1|n-1) \tag{4.116}$$

$$\mathbf{K}(n, n-1) = \mathbf{K}(n-1). \tag{4.117}$$

	T_p	T_e	T_a	α	ϵ	$\alpha(1\|0)$	$\epsilon(1\|0)$
Ex1	0.7983	0.8448	0.0690	10.9179	11.2937	11	12
Ex2	0.7607	0.8036	0.0536	23.9384	21.0114	24	21
Ex3	0.8298	0.8696	0.0797	18.1079	9.5485	19	9
Ex4	0.7018	0.7345	0.0655	12.9454	16.4931	13	16.5

Table 4.2 Estimated LF model parameters. Ex1, Ex2, Ex3 correspond to the extracted pluck excitations (1), (2), (3) in Fig. 4.26 and Ex4 corresponds to the extracted pluck excitation (5) in Fig. 4.27.

In Table 4.2, estimated LF model parameters are shown for pluck excitations that are extracted using the RLS algorithm discussed in the previous chapters. Figure 4.29 depicts the time-domain signal and frequency magnitude response of an extracted excitation and the model derived using the EKF. In the time domain, the LF model well approximates the region where the acceleration excitation is returning to zero from the minimum point, corresponding to the return phase of GFD; but it appears that the sharp rising at the beginning of the extracted excitation is not modeled as well as the return phase. However, synthesized plucking sounds using the synthesized excitation actually sound quite natural. Sound examples are available[3].

[3]http://www.music.mcgill.ca/∼lee/pluckexcitation

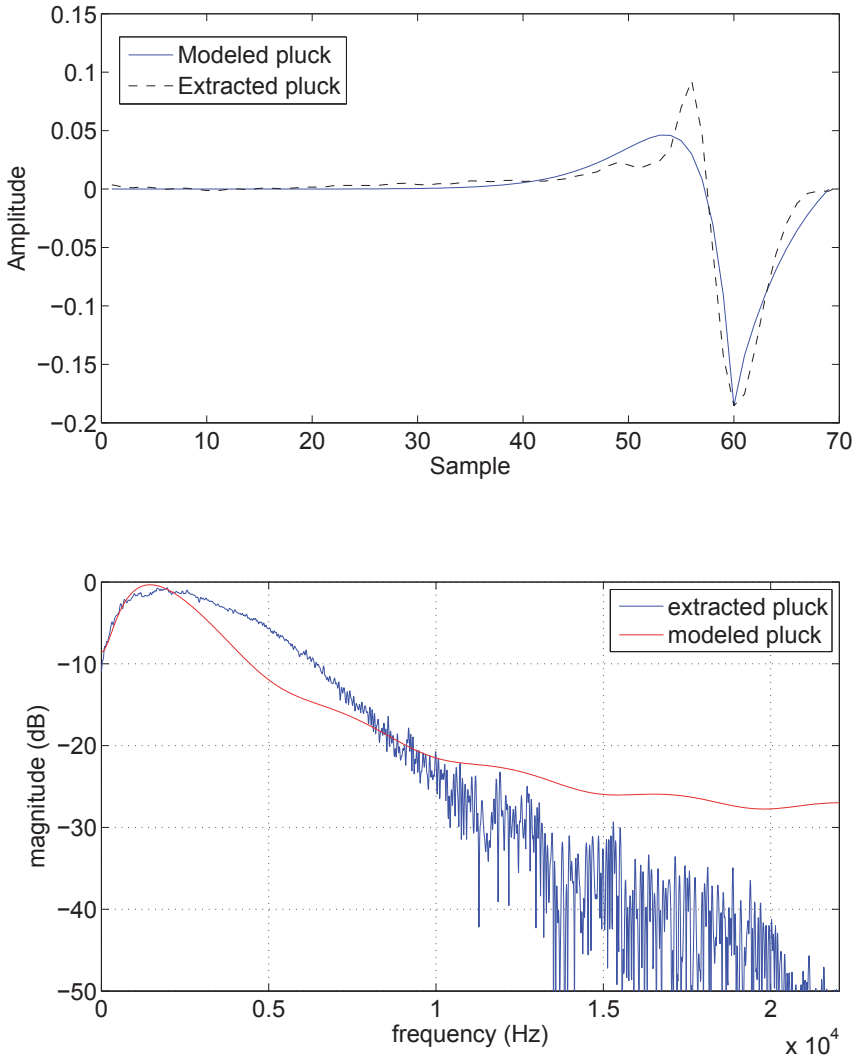

Fig. 4.29 Extracted excitation and modeled excitation. Top pane illustrates the extracted excitation and the modeled excitation in the time domain. $\alpha(1|0) = 18$, $\epsilon(1|0) = 9$. The bottom pane illustrates the magnitude responses of the extracted and modeled excitations.

4.10 Discussion - Finger/String Interaction

So far we have assumed the overall plucked-string sound generation mechanism as a linear time invariant (LTI) system, as many physically-based pluck sound synthesis techniques do. However, rigorously, when plucking action takes place, there should be a bi-lateral interaction between a string and a finger/plectrum. Here we investigate how the interaction model can be associated with the extraction and modeling of pluck excitations.

4.10.1 Finger/plectrum model

In [79], Cuzzucoli and Lombardo proposed a physical finger/plectrum model based on lumped elements (masses, springs and dampers) that is integrated with the DW structure. Evangelista and Eckerholm [3] and, later, Evangelista and Smith [19] enhanced Cuzzucoli and Lombardo's model. In their model the interaction between the finger/plectrum during the plucking action at point x_p on the string is described. These models are rooted in the following equation of motion,

$$(M + \mu\Delta)\frac{\partial^2 y}{\partial t^2} + R\frac{\partial y}{\partial t} + Ku - f(t) = f_0(t) \tag{4.118}$$

where y is the displacement of the string. M, K and R are the mass, stiffness, and damping parameters of the finger. μ and Δ are the linear mass density (kg/m) of the string and the length of the string segment, centered at x_p. $f_0(t)$ is the force that the finger/plectrum exerts on the string. The force $f(t)$ is a transverse tensile force acting on the string segment. With the assumption of

a small deformation of the string, $f(t)$ is given as:

$$f(t) = K_0 \left(\left. \frac{\partial y}{\partial x} \right|_{x_p + \frac{\Delta}{2}} - \left. \frac{\partial y}{\partial x} \right|_{x_p - \frac{\Delta}{2}} \right) \tag{4.119}$$

where K_0 is the tension of the string. Assuming $\Delta = X$ (X: the spatial sampling interval) and by substituting Eq. 4.119 into Eq. 4.118 and applying the finite centered difference scheme, we can obtain the scattering junction structure in a DW illustrated in Fig. 4.30 (details about this can be found in [3]). The scattering junction relation can be described in matrix form [3][19] as

$$\begin{bmatrix} Y_{out}^{-}(z) \\ Y_{out}^{+}(z) \end{bmatrix} = \mathbf{S}(z) \begin{bmatrix} Y_{in}^{-}(z) \\ Y_{in}^{+}(z) \end{bmatrix} + \frac{C(z)F_0(z)}{2} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \tag{4.120}$$

where $Y_{in/out}^{\pm}(z)$ are the z-transforms of the signals $y_{in/out}^{\pm}(n)$ and the matrix $\mathbf{S}(z)$ is given as:

$$\mathbf{S}(z) = \frac{1}{2} \begin{bmatrix} Q(z)+1 & Q(z)-1 \\ Q(z)-1 & Q(z)+1 \end{bmatrix} \tag{4.121}$$

In the model proposed by Evangelista and Eckerholm [3], the transfer function $Q(z)$ is given as:

$$Q(z) = \frac{1}{B(z)} \tag{4.122}$$

$$B(z) = \frac{M}{\mu X}(1 - z^{-1})^2 + \rho(1 - z^{-2}) + \kappa z^{-1} + 1 \tag{4.123}$$

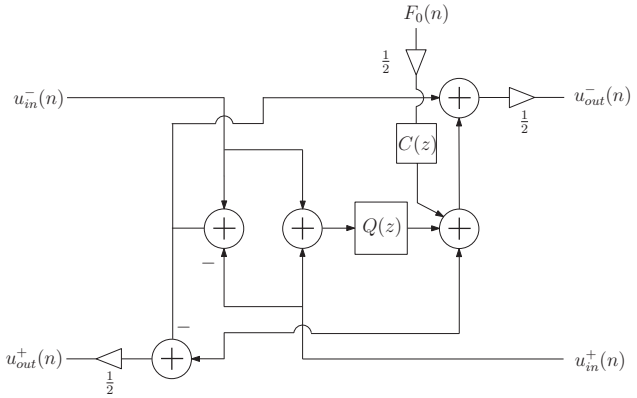

Fig. 4.30 Diagram of the plucking scattering junction, from [3] (slightly modified).

and

$$\rho = \frac{R}{2\sqrt{\mu K_0}}, \qquad \kappa = \frac{KX}{K_0} \tag{4.124}$$

The force coupling filter $C(z)$ is

$$C(z) = \frac{2Xz^{-1}}{K_0 B(z)}. \tag{4.125}$$

The model defined with the transfer functions above is referred to as the CLEE (Cuzucolli-Lombardo-Evangelista-Eckerholm) model. In [19], the CLEE model is enhanced in such a way that the passivity of $\mathbf{S}(z)$ is guaranteed. For this model, the equation of motion (Eq. 4.118) is Laplace-transformed so that [19]:

$$[(M + \mu\Delta)s^2 + Rs + K]Y(x, s) = F(s) + F_0(s) \tag{4.126}$$

$$F(s) = K_0 \left(\frac{\partial Y(x, s)}{\partial x}\bigg|_{x+\frac{\Delta}{2}} - \frac{\partial Y(x, s)}{\partial x}\bigg|_{x-\frac{\Delta}{2}} \right) \tag{4.127}$$

where $Y(x, s)$ is the Laplace transform of $y(x, t)$ with respect to time. Also, $f(t)$ is transformed to $F(s)$ and the Laplace transforms of the incoming and

outgoing waves are

$$Y^-(x,s) = \mathcal{L}[y^-(x,t)](s) = \mathcal{L}[y_l(t+x/c)](s) = e^{+\frac{sx}{c}}Y_l(s) \quad (4.128)$$

$$Y^+(x,s) = \mathcal{L}[y^+(x,t)](s) = \mathcal{L}[y_r(t-x/c)](s) = e^{-\frac{sx}{c}}Y_r(s) \quad (4.129)$$

where $\mathcal{L}(\cdot)$ is the Laplace transform operator, and the derivatives with respect to x are

$$\frac{\partial Y^-(x,s)}{\partial x} = +\frac{s}{c}e^{+\frac{sx}{c}}Y_l(s) = +\frac{s}{c}Y^-(x,s) \quad (4.130)$$

$$\frac{\partial Y^+(x,s)}{\partial x} = -\frac{s}{c}e^{-\frac{sx}{c}}Y_r(s) = -\frac{s}{c}Y^+(x,s). \quad (4.131)$$

Then Eq. 4.127 can be written as

$$F(s) = \frac{K_0 s}{c}[Y^-(x_p+\tfrac{\Delta}{2},s) - Y^+(x_p+\tfrac{\Delta}{2},s) - Y^-(x_p-\tfrac{\Delta}{2},s) + Y^+(x_p-\tfrac{\Delta}{2},s)]. \quad (4.132)$$

If we substitute Eq. 4.130 and Eq. 4.131 into Eq. 4.126, we get

$$\begin{aligned}(Y^-(x_p-\tfrac{\Delta}{2},s) + Y^+(x_p-\tfrac{\Delta}{2},s))E(s) - F(s) &= F_0(s)\\ (Y^-(x_p+\tfrac{\Delta}{2},s) + Y^+(x_p+\tfrac{\Delta}{2},s))E(s) - F(s) &= F_0(s)\end{aligned} \quad (4.133)$$

where

$$E(s) = (M+\mu\Delta)s^2 + Rs + K. \quad (4.134)$$

By substituting Eq. 4.132 into Eq. 4.133, we obtain

$$\begin{bmatrix} Y^-(x_p - \frac{\Delta}{2}, s) \\ Y^+(x_p + \frac{\Delta}{2}, s) \end{bmatrix} = \tilde{\mathbf{S}}(s) \begin{bmatrix} Y^-(x_p + \frac{\Delta}{2}, s) \\ Y^+(x_p - \frac{\Delta}{2}, s) \end{bmatrix} + \frac{\tilde{C}(s)F_0(s)}{2} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad (4.135)$$

where $\tilde{\mathbf{S}}(s)$ is

$$\tilde{\mathbf{S}}(s) = \frac{1}{2} \begin{bmatrix} \tilde{Q}(s) + 1 & \tilde{Q}(s) - 1 \\ \tilde{Q}(s) - 1 & \tilde{Q}(s) + 1 \end{bmatrix}$$

and $\tilde{Q}(s)$, $\tilde{C}(s)$ are

$$\tilde{Q}(s) = -\frac{cE(s) - 2sK_0}{cE(s) + 2sK_0} \quad (4.136)$$

$$\tilde{C}(s) = \frac{2c}{cE(s) + 2sK_0}. \quad (4.137)$$

Applying the bilinear transformation

$$s = \frac{2}{T}\frac{z-1}{z+1}. \quad (4.138)$$

we finally get the z-domain transfer functions $Q(z)$, $C(z)$ as

$$Q(z) = \tilde{Q}\left(\frac{2}{T}\frac{z-1}{z+1}\right) \quad (4.139)$$

$$C(z) = \tilde{C}\left(\frac{2}{T}\frac{z-1}{z+1}\right) \quad (4.140)$$

4.10.2 Finger/plectrum-String Interaction with SDL

We can now investigate how the interaction model described thus far is represented within the SDL model framework. To this end, we re-write the scat-

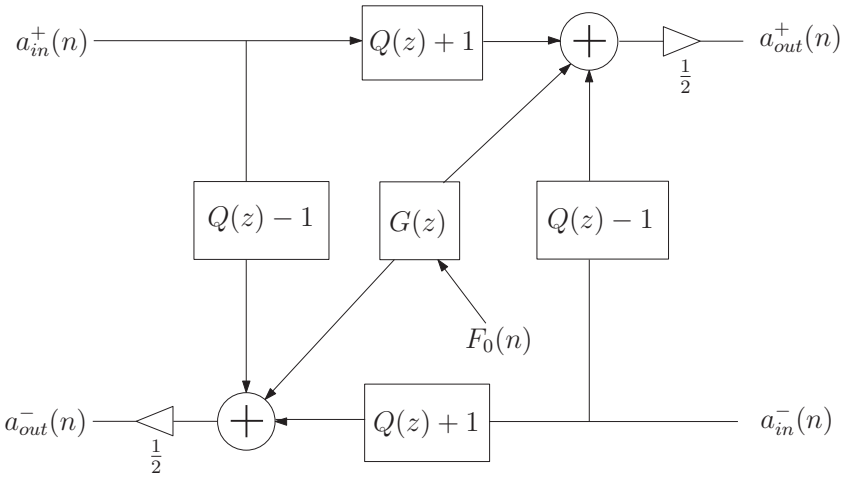

Fig. 4.31 Scattering junction at the excitation point.

tering junction that represents the interaction (Eq. 4.135) as follows:

$$\begin{bmatrix} Y^-_{out}(z) \\ Y^+_{out}(z) \end{bmatrix} = \frac{1}{2} \begin{bmatrix} Q(z)+1 & Q(z)-1 \\ Q(z)-1 & Q(z)+1 \end{bmatrix} \begin{bmatrix} Y^-_{in}(z) \\ Y^+_{in}(z) \end{bmatrix} + \frac{C(z)F_0(z)}{2} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad (4.141)$$

where $Y^+_{out}(z)$, $Y^+_{in}(z)$, $Y^-_{out}(z)$ and $Y^-_{in}(z)$ are the z-transforms of string displacements $y^+_{out}(n)$, $y^+_{in}(n)$, $y^-_{out}(n)$ and $y^-_{in}(n)$ as shown in Fig. 4.31. We can write down the formula above describing finger/string interaction in terms of z domain representations of acceleration wave components as follows,

$$\begin{bmatrix} A^-_{out}(z) \\ A^+_{out}(z) \end{bmatrix} = \frac{1}{2} \begin{bmatrix} Q(z)+1 & Q(z)-1 \\ Q(z)-1 & Q(z)+1 \end{bmatrix} \begin{bmatrix} A^-_{in}(z) \\ A^+_{in}(z) \end{bmatrix} + \frac{I^2(z)G(z)F_0(z)}{2} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad (4.142)$$

where $A^+_{out}(z)$, $A^+_{in}(z)$, $A^-_{out}(z)$ and $A^-_{in}(z)$ are the z-transforms of acceleration wave components $a^+_{out}(n)$, $a^+_{in}(n)$, $a^-_{out}(n)$ and $a^-_{in}(n)$. $I(z)$ is an integrator. By

letting $Q'(z) = (Q(z) - 1)/2$, we get

$$\begin{bmatrix} A^-_{out}(z) \\ A^+_{out}(z) \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 2 + 2Q'(z) & 2Q'(z) \\ 2Q'(z) & 2 + 2Q'(z) \end{bmatrix} \begin{bmatrix} A^-_{in}(z) \\ A^+_{in}(z) \end{bmatrix} + \frac{I^2(z)G(z)F_0(z)}{2} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \tag{4.143}$$

Let us first consider a non-coupled case in which $P(z) = 1$ so that the scattering matrix is given as the following identity matrix:

$$\begin{bmatrix} A^-_{out}(z) \\ A^+_{out}(z) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} A^-_{in}(z) \\ A^+_{in}(z) \end{bmatrix} + \frac{I^2(z)G(z)F_0(z)}{2} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \tag{4.144}$$

$$A^-_{out}(z) = A^-_{in}(z) + \frac{I^2(z)G(z)F_0(z)}{2} \tag{4.145}$$

$$A^+_{out}(z) = A^+_{in}(z) + \frac{I^2(z)G(z)F_0(z)}{2} \tag{4.146}$$

According to our DW model of the plucked string in Fig. 4.12, we have the relation

$$A^+_{in}(z) = H_{E2,E1}(z)A^-_{out}(z). \tag{4.147}$$

Using Eqs. 4.145 - 4.147, we also have

$$A^+_{out}(z) = H_{E2,E1}(z)(A^-_{in}(z) + \frac{I^2(z)G(z)F_0(z)}{2}) + \frac{I^2(z)G(z)F_0(z)}{2} \quad (4.148)$$

$$A^-_{in}(z) = H^2_{E1,O1}(z)H_{O1,O2}(z)A^+_{out}(z) \quad (4.149)$$

$$A_{1,N_{pu}}(z) = H_{E1,O1}(z)A^+_{out}(z) \quad (4.150)$$

$$A_{2,N_{pu}}(z) = H^{-1}_{E1,O1}(z)A^-_{in}(z) = H_{E1,O1}(z)H_{O1,O2}(z)A^+_{out}(z) \quad (4.151)$$

where $A_{1,N_{pu}}(z)$, $A_{2,N_{pu}}(z)$ are the z-transforms of $a_{1,N_{pu}}(n)$, $a_{2,N_{pu}}(n)$, respectively. Inserting Eq. 4.149 into Eq. 4.148 gives

$$(1 - H_{E2,E1}(z)H^2_{E1,O1}(z)H_{O1,O2}(z))A^+_{out}(z) = \frac{H_{E2,E1}(z)I^2(z)G(z)F_0(z)}{2} + \frac{I^2(z)G(z)F_0(z)}{2} \quad (4.152)$$

Considering the coupled case, from Eq. 4.143 we have,

$$A^-_{out}(z) = (1 + Q'(z))A^-_{in}(z) + Q'(z)A^+_{in}(z) + \frac{I^2(z)G(z)F_0(z)}{2} \quad (4.153)$$

$$A^+_{out}(z) = Q'(z)A^-_{in}(z) + (1 + Q'(z))A^+_{in}(z) + \frac{I^2(z)G(z)F_0(z)}{2} \quad (4.154)$$

and using Eqs. 4.145 - 4.147, we get

$$A^+_{out}(z) = Q'(z)A^-_{in}(z) + \frac{H_{E2,E1}(z)(1 + Q'(z))^2}{1 - H_{E2,E1}(z)Q'(z)}A^-_{in}(z) + \frac{H_{E2,E1}(z)(1 + Q'(z))I^2(z)G(z)F_0(z)}{2(1 - H_{E2,E1}(z)Q'(z))} + \frac{I^2(z)G(z)F_0(z)}{2}. \quad (4.155)$$

Substituting Eq. 4.149 into Eq. 4.155, the results are

$$\begin{aligned} A^+_{out}(z) &= Q'(z)H^2_{E1,O1}(z)H_{O1,O2}(z)A^+_{out}(z) \\ &+ \frac{H_{E2,E1}(z)(1+Q'(z))^2}{1-H_{E2,E1}(z)Q'(z)}H^2_{E1,O1}(z)H_{O1,O2}(z)A^+_{out}(z) \\ &+ \frac{H_{E2,E1}(z)(1+Q'(z))I^2(z)G(z)F_0(z)}{2(1-H_{E2,E1}(z)Q'(z))} + \frac{I^2(z)G(z)F_0(z)}{2} \end{aligned} \tag{4.156}$$

and

$$\begin{aligned} &(1 - H_{E2,E1}(z)H^2_{E1,O1}(z)H_{O1,O2}(z) \\ &\quad - H_{E2,E1}(z)Q'(z) - Q'(z)H^2_{E1,O1}(z)H_{O1,O2}(z) \\ &\quad - 2H_{E2,E1}(z)Q'(z)H^2_{E1,O1}(z)H_{O1,O2}(z))A^+_{out}(z) \\ &\quad = \frac{H_{E2,E1}(z)I^2(z)G(z)F_0(z)}{2} + \frac{I^2(z)G(z)F_0(z)}{2} \end{aligned} \tag{4.157}$$

Let $C(z)$ denote the coupling term in Eq. 4.157 in a way that,

$$\begin{aligned} C(z) &= -H_{E2,E1}(z)Q'(z) - Q'(z)H^2_{E1,O1}(z)H_{O1,O2}(z) \\ &\quad - 2H_{E2,E1}(z)Q'(z)H^2_{E1,O1}(z)H_{O1,O2}(z) \end{aligned} \tag{4.158}$$

$$= -Q'(z)(H_{E2,E1}(z) + H^2_{E1,O1}(z)H_{O1,O2}(z) + 2H_{E2,E1}(z)H^2_{E1,O1}(z)H_{O1,O2}(z)) \tag{4.159}$$

then Eq. 4.157 can be re-written as

$$\begin{aligned} &(1 - H_{E2,E1}(z)H^2_{E1,O1}(z)H_{O1,O2}(z) + C(z))A^+_{out}(z) \\ &= (1 - H_{loop}(z) + C(z))A^+_{out}(z) \\ &= \frac{H_{E2,E1}(z)I^2(z)G(z)F_0(z)}{2} + \frac{I^2(z)G(z)F_0(z)}{2} \end{aligned} \tag{4.160}$$

where $H_{loop}(z) = H_{E2,E1}(z)H^2_{E1,O1}(z)H_{O1,O2}(z)$ as defined in Eq. 4.24. The observation $A^+_{out}(z)$ can then be written as

$$A^+_{out}(z) = \frac{1 + H_{E2,E1}(z)}{1 - H_{loop}(z)} \frac{I^2(z)G(z)F_0(z)}{2} \quad (non - coupling) \quad (4.161)$$

$$A^+_{out}(z) = \frac{1 + H_{E2,E1}(z)}{1 - H_{loop}(z) + C(z)} \frac{I^2(z)G(z)F_0(z)}{2} \quad (coupling) \quad (4.162)$$

Using, either Eq. 4.161 or Eq. 4.162 and Eq. 4.150, Eq. 4.151, we have,

$$\begin{aligned} A_{N_{pu}}(z) &= A_{1,N_{pu}}(z) + A_{2,N_{pu}}(z) \\ &= \frac{H^2_{E1,O1}(z)(1 + H_{O1,O2}(z))(1 + H_{E2,E1}(z))}{1 - H_{loop}(z) + C(z)} \frac{I^2(z)G(z)F_0(z)}{2} \end{aligned} \quad (4.163)$$

If we assume the non-coupling case and perform inverse filtering accordingly, then we get

$$\tilde{F}(z) = \frac{1 - H_{loop}(z)}{1 - H_{loop}(z)} \frac{I^2(z)G(z)F_0(z)}{2} = \frac{I^2(z)G(z)F_0(z)}{2} \quad (4.164)$$

which shows that the result of the inverse-filtering $\tilde{F}(z)$ is the same as the modeled excitation. If we assume the coupling case, then

$$\tilde{F}(z) = \frac{1 - H_{loop}(z)}{1 - H_{loop}(z) + C(z)} \frac{I^2(z)G(z)F_0(z)}{2} \quad (4.165)$$

Using the series expansion, the equation above becomes,

$$\tilde{F}(z) \sim (1 + \frac{C(z)}{1 - H_{loop}(z)} + \cdots) \frac{I^2(z)G(z)F_0(z)}{2} \quad (4.166)$$

As seen in Eq. 4.166, if we assume that the bilateral interaction exists, the inverse-filtering based on the SDL model would not be appropriate to extract

the input excitation since the coupling term $C(z)$ includes the parameters associated with the finger model. In addition, as the input excitation is characterized by both $G(z)$ and $F_0(z)$, deconvolution of these two terms would be another issue if one were to aim at estimating the finger model parameters in $G(z)$. In order to solve these problems, the plucked string model should be viewed within another framework in which the coupling term could be taken into account more robustly. This will remain for future work.

4.11 Conclusion

In this chapter, we have proposed an intuitive method to simply extract the pluck excitation from a plucked-string signal based on time windowing. It was inspired by observing the way traveling wave components behave in the plucked-string sound signal and by comparison with a DW simulation. Inspired by this time-windowing method, a pluck excitation extraction technique based on inverse-filtering associated with the SDL model is also proposed. We found that pluck excitations extracted using the time-windowing method and the inverse-filtering based method are well matched in certain cases. The inverse-filtering based method is appropriate in most cases of pluck excitations, whereas the time-windowing based method has its limit. The excitation extracted by the proposed methods is compact and physically more meaningful, facilitating the use of excitations for synthesis in conjunction with physical models such as DW and SDL. In addition, we carried out a research on constructing a parametric model of excitations and estimating the associated parameters.

All the tasks described in this chapter are attempts to investigate the sources that are used to synthesize the sounds of musical instruments. Con-

trary to the sources used for generating abstract, non-musical sounds investigated in the previous chapters, the sources associated with musical instrument sounds have much to do with the expression intended by performers. A plucked-string sound, as one of the simplest and the most intuitive cases in terms of sound generation mechanism, and the source-filter-based extraction of pluck excitations obtained from this research will provide more chances for investigating performance expressions quantitatively and, accordingly, lead to richer flexibility in synthesis. In addition to the future work derived from the above discussion, there is the chance for enhancement of various aspects of the work in this chapter. First, we could take into account a typical dual-polarization of a string vibration to build either a DW or an SDL model. This would enable one to investigate the angle of plucking more systematically. Another issue worth considering involves the possibility of constructing a better parametric model of excitations. In this task, though the LF model originally developed for GFD can be used, a better model customized to pluck excitations could be constructed. Furthermore, other kinds of estimation techniques besides the EKF might be tried, and they could possibly yield better and more interesting results.

Chapter 5

Conclusions

In this thesis, various source types used for sound analysis/re-synthesis are thoroughly investigated and novel analysis/re-synthesis methods are proposed. Sources are defined in the context of a source-filter model and a granular analysis/re-synthesis framework. Particular cases are selected to verify our viewpoint regarding source types, and novel analysis/re-synthesis algorithms are proposed for each case.

First, an analysis and synthesis scheme of rolling sounds is proposed. Based on the contact timing information obtained through a process similar to onset detection, an overall rolling sound is segmented into individual contact events and fed into an analysis/synthesis system for the estimation of time-varying filters. Subband-LP analysis allows greater focusing on significant spectral features. For resynthesis, synthesized contact events are concatenated to create the final rolling sound. It is found that the proposed scheme works better for specific kinds of rolling sounds in which each micro contact is relatively well preserved. This novel 'divide and conquer' approach allows for analysis and synthesis of rolling sounds at a single contact level, linking spatially varying resonance/anti-resonance characteristics of rolling phenomena to the

temporal interpretation based on time-varying filter models. This spatial-time correspondence improves existing approaches of applying source/filter models to analysis/synthesis of rolling sounds with the capability of taking the varying resonance/anti-resonance into account more systematically.

Next, a novel granular analysis/synthesis scheme for complex sounds is proposed. The granular analysis component segments a given sound into grains by detecting transient events. Through the analysis to distinguish stationary/non-stationary regions in the sound, different segmentation parameters can be assigned for each region, allowing the user to apply different criteria for defining the grain in each region. Furthermore, several useful audio features are extracted from each grain for potential use in synthesis. With the granular synthesis component, the user can synthesize sounds with pre-analyzed grains. In addition, various kinds of time modification are possible for flexible synthesis with convincing sound quality. Both granular analysis and synthesis components are provided with GUIs. The proposed granular analysis/synthesis scheme differentiates itself from others in both analysis and synthesis. In the analysis stage, the proposed scheme is capable of flexible parameter adjustments with respect to the characteristics of given sounds. In this respect, we also proposed a novel criteria for grain segmentation referred to as the 'stationarity measure' that can classify given sounds based on how consistent and regular the nature of the sound is, allowing for different parameter settings within a sound. The time modification schemes proposed for the synthesis stage includes novel strategies, the grain extension method and the additional grain-based method, for efficiently filling unnecessary gaps caused from time stretching and shrinking. In addition, the grain time remapping enriches the flexibility of the synthesis in conjunction with time stretch-

ing/shrinking, allowing users to rearrange the temporal orders of grains at will via a graphic interface.

Finally, we propose a simple but physically intuitive method to extract the pluck excitation from a plucked string signal using time windowing and another physically informed method based on inverse-filtering associated with the physical model of plucked strings. Both methods are well matched in certain cases; however, the inverse-filtering method is applicable to a wider range of cases of pluck excitations than the time-windowing method. The excitation extracted by the proposed methods is compact in time and physically meaningful, so it can be directly used with physical models for plucked-string sound synthesis. In addition, a parametric model of excitations based on the LF model is proposed. We expect that pluck excitations obtained using the proposed methods will contribute in a quantitative way to research on performance expressions. The major contribution of the research is to find a way to extract the 'temporally-meaningful and accurate' pluck excitation using a parametric physical model. The extracted pluck excitation clearly defines the correspondence between the excitation in the acoustic domain and the signal domain. As the pluck excitation extracted by the proposed algorithm preserves the temporal evolution, we are able to investigate the temporal behavior of pluck excitations not only in the physics domain but also in the signal domain. Also, we demonstrated the use of a parametric model for pluck excitation inspired by speech synthesis.

We believe that the specified and categorized source components developed through this research will contribute to the evolution of sound synthesis techniques used in computer-based music by providing more and greater flexibility.

5.1 Future Work

Future work will include several research tasks that could potentially enhance the current research outcomes. One would be finding a clever way for grain compression other than using the 'Offset Threshold' parameter for the proposed granular analysis/synthesis scheme. In general, it is likely that redundant grains exist in a dictionary, and they incur unnecessary consumption of computer resources. By clustering redundant grains through the use of a proper machine learning technique, the size of a dictionary can be reduced while the quality of sound synthesis is maintained. Another problem to think about is how to figure out the inherent rhythmic aspect of a given sound. In contrast to music or speech, environmental sounds are quite often non-rhythmic or have rhythms that are hard to analyze (e.g. the sound of applause). However, if we could analyze the rhythm of a sound, it would be beneficial insofar as it would broaden the flexibility of the synthesis system. There are some issues remaining as future work regarding the research of pluck excitation extraction. First, we could take into account a typical dual-polarization of a string vibration to build either a DW or an SDL model. This would enable one to investigate the angle of plucking more systematically. Another issue worth considering involves the possibility of constructing a better parametric model of excitations. In this task, though the LF model originally developed for GFD can be used, a better model customized to pluck excitations could be constructed. Furthermore, other kinds of estimation techniques besides the EKF might be tried, and they could possibly yield better and more interesting results.

Future work should also include perceptual studies to better inform the results of all the synthesis approaches discussed in this thesis. Necessary per-

ceptual studies would involve not only the comparison between the synthesized sound and the original sound and but also the evaluation of how natural or realistic the synthesized sounds are without comparison to the original sounds, in order to verify and evaluate the proposed analysis/synthesis schemes.

References

[1] M. Slaney, "Auditory toolbox, version 2," Tech. Rep. #1998-010, Interval Research Corporation.

[2] J. O. Smith, *Physical Audio Signal Processing*. W3K Publishing, 2007.

[3] G. Evangelista and F. Eckerholm, "Player-instrument interaction models for digital waveguide synthesis of guitar: Touch and collisions," *IEEE transactions on Audio, Speech, and Language Processing*, vol. 18, no. 4, pp. 822–832, 2010.

[4] P. R. Cook, "Modeling bill's gait: analysis and parametric synthesis of walking sounds," in *Proc. of AES 22nd International Conference on Virtual, Synthetic and Entertainment Audio*, (Espoo, Finland), pp. 73–78, February 2002.

[5] C. E. L. Peltola and P. R. Cook, "Synthesis of hand clapping sounds," *IEEE Transactions of Audio, Speech and Language Signal Processing*, vol. 15, no. 3, pp. 1021–1097, 2007.

[6] K. V. den Doel, P. G. Kry, and D. K. Pai, "Foleyautomatic:physically based sound effects for interactive simulation and animation," in *Proc. of International Conference of Computer Graphics and Interactive Techniques (SIGGRAPH)*, 2001.

[7] M. Rath, "An expressive real-time sound model of rolling," in *the 6th International Conference on Digital Audio Effects (DAFx-03)*, (London, UK), Sept. 2003.

[8] D. Rocchesso and F. Fontana, *The Sounding Object*. Edizioni di Mondo Estremo, 2003.

[9] M. Lagrange, G. P. Scavone, and P. Depalle, "Analysis/synthesis of sounds generated by sustained contact between rigid objects," *IEEE Transactions of Audio, Speech and Language Signal Processing*, vol. 18, no. 3, pp. 509–518, 2010.

[10] D. Schwarz, "Corpus-based concatenative synthesis," *IEEE signal processing magazine*, vol. 24, no. 2, pp. 92–104, 2007.

[11] C. Picard, N. Tsingos, and F. Faure, "Retargetting example sounds to interactive physics-driven animations," in *Proc. of AES 35th International Conference: Audio for Games*, (London, UK), February 2009.

[12] J. Laroche and J. L. Meillier, "Multichannel excitation/filter modeling for percussive sounds with application to the piano," *IEEE Transactions on Speech and Audio Processing*, vol. 2, no. 2, pp. 329–344, 1994.

[13] V. Välimäki, J. Huopaniemi, and Z. Jánosy, "Physical modeling of plucked-string instruments with application to real-time sound synthesis," *Journal of Audio Engineering Society*, vol. 44, no. 5, pp. 331–353, 1996.

[14] T. Tolonen, "Model-based analysis and resynthesis of acoustic guitar tones," Master's thesis, Helsinki University of Technology, Espoo, Finland, 1998.

[15] V. Välimäki and T. Tolonen, "Development and calibration of a guitar synthesizer," *Journal of Audio Engineering Society*, vol. 46, no. 9, pp. 766–777, 1998.

[16] N. Lee, Z. Duan, and J. O. Smith, "Excitation signal extraction for guitar tones," in *Proc. of International Computer Music Conference (ICMC-07)*, (Copenhagen, Denmark), 2007.

[17] J. Lee, P. Depalle, and G. Scavone, "Analysis/synthesis of rolling sounds using a source-filter approach," in *Proc. of International Conference on Digital Audio Effects (DAFx-10)*, (Graz, Austria), 2010.

[18] J. Lee, P. Depalle, and G. Scavone, "On the extraction of excitation from a plucked string sound in time domain," *Canadian Acoustics*, vol. 39, no. 2, pp. 126–127, 2011.

[19] G. Evangelista and J. O. Smith, "Structurally passive scattering element for modelling guitar pluck action," in *Proc. of International Conference on Digital Audio Effects (DAFx-10)*, (Graz, Austria), Sept. 2010.

[20] K. V. den Doel, P. G. Kry, and D. K. Pai, "Foleyautomatic : physically based sound effects for interactive simulation and animation," in *Proc. of International Conference of Computer Graphics and Interactive Techniques (SIGGRAPH)*, 2001.

[21] K. Hunt and F. R. E. Crossley, "Coefficient of restitution interpreted as damping in vibroimpact," *Journal of Applied Mechanics*, vol. 42, pp. 440–445, June 1975.

[22] C. Stoelinga and A. Chaigne, "Time-domain modeling and simulation of rolling objects," *Acta Acustica united with Acustica*, vol. 93, no. 2, pp. 290–304, 2007.

[23] A. Chaigne and C. Lambourg, "Time-domain simulation of damped impacted plates. i. theory and experiments," *J. Acoust. Soc. Am.*, vol. 109, no. 4, pp. 1422–1432, 2001.

[24] J. P. Bello, L. Daudet, S. Abdallah, C. Duxbury, M. Davies, and M. B. Sandler, "A tutorial on onset detection in music signals," *IEEE Transactions of speech and audio processing*, vol. 13, no. 5, pp. 1035–1047, 2005.

[25] P. P. Vaidyanathan, *Multirate systems and filter banks*. Prentice Hall, 1993.

[26] T. Q. Nguyen and P. P. Vaidyanathan, "Two-channel perfect reconstruction FIR QMF structures which yield linear-phase analysis and synthesis filters," *IEEE Trans. Acoustics, Speech and Signal Processing*, vol. 37, pp. 676–690, May 1989.

[27] S. M. Kay and S. L. Marple Jr, "Spectrum analysis : a modern perspective," *Proceedings of the IEEE*, vol. 69, pp. 1380–1419, May 1981.

[28] D. B. Rao and S. Y. Kung, "Adaptive notch filtering for the retrieval of sinusoids in noise," *IEEE Trans. Acoust., Speech and Signal Processing*, vol. 32, no. 4, pp. 791–802, 1984.

[29] J. Makhoul, "Linear prediction: A tutorial review," *Proceedings of the IEEE*, vol. 63, no. 4, pp. 561–581, 1975.

[30] D. Gabor, "Acoustical quanta and the theory of hearing," *Nature*, vol. 159, pp. 591–594, May 1947.

[31] I. Xenakis, *Formalized Music*. Indiana University Press, 1971.

[32] C. Roads, "Introduction to granular synthesis," *Computer Music Journal*, vol. 12, no. 2, pp. 27–34, 1988.

[33] B. Truax, "Real-time granular synthesis with a digital signal processor," *Computer Music Journal*, vol. 12, no. 2, pp. 14–26, 1988.

[34] A. Hunt and A. Black, "Unit selection in a concatenative speech synthesis system using a large speech database," in *Proc. of International Conference on Acoustics, Speech and Signal Processing (ICASSP 96)*, (Atlanta, Georgia), pp. 373–376, May 1996.

[35] D. Schwarz, "Concatenative sound synthesis : The early years," *Journal of New Music Research*, vol. 35, no. 1, pp. 3–22, 2006.

[36] L. Lu, L. Wenyin, and H.-J. Zhang, "Audio textures : theory and applications," *IEEE Transactions of speech and audio processing*, vol. 12, no. 2, pp. 156–167, 2004.

[37] Y. Dobashi, T. Yamamoto, and T. Nishita, "Synthesizing sound from turbulent field using sound textures for interactive fluid simulation," in *Proc. of Eurographics*, pp. 539–546, 2004.

[38] R. Hoskinson and D. K. Pai, "Manipulation and resynthesis with natural grains," in *Proc. of the International Computer Music Conference (ICMC 01)*, (San Francisco, U.S.A.), pp. 338–341, 2001.

[39] A. Lazier and P. R. Cook, "MOSIEVIUS: Feature driven interactive audio mosaicing," in *Proc. of the International Conference on Digital Audio Effects (DAFx-03)*, (London, U.K.), pp. 323–326, Sept. 2003.

[40] B. L. Sturm, "MATCONCAT : an application for exploring concatenative sound synthesis using matlab," in *Proc. of the International Conference on Digital Audio Effects (DAFx-04)*, (Naples, Italy), pp. 323–326, Oct. 2004.

[41] D. Schwarz, R. Cahen, and S. Britton, "Principles and applications of interactive corpus-based concatenative synthesis," in *Journées d'Informatique Musicale (JIM)*, (GMEA, Albi, France), March 2008.

[42] D. López, F. Martí, and E. Resina, "Vocem: An application for real-time granular synthesis," in *Proc. of International Conference on Digital Audio Effects (DAFx-98)*, 1998.

[43] A. V. Oppenheim and R. W. Schafer, *Discrete-time signal processing*. Prentice-Hall, second ed., 1998.

[44] X. Serra, *A system for sound analysis/transformation/synthesis based on a deterministic plus stochastic decomposition*. PhD thesis, CCRMA, Stanford University, Stanford, CA, 1989.

[45] J. D. Johnston, "Transform coding of audio signals using perceptual noise criteria," *IEEE journal on selected areas in communication*, vol. 6, no. 2, 1988.

[46] I. Kauppinen and K. Roth, "Audio signal extrapolation : Theory and applications," in *Proc. of the 5th Int. Conf. on Digital Audio Effects (DAFx-02)*, (Hamburg, Germany), September 2002.

[47] W. Etter, "Restoration of a discrete-time signal segment by interpolation based on the left-sided and right-sided autoregressive parameters," *IEEE transactions on signal processing*, vol. 44, no. 5, pp. 1124–1135, 1996.

[48] R. C. Maher, "A method of extrapolation of missing digital audio data," *Journal of Audio Engineering Society*, vol. 42, no. 12, pp. 350–357, 1994.

[49] F. Itakura and S. Saito, "An analysis-synthesis telephony based on the maximum likelihood method," in *Proc. of 6th International Congress of Acoustics*, (Tokyo, Japan), 1968.

[50] S. Sigurdsson, K. B. Petersen, and T. Lehn-Schiøler, "Mel frequency cepstral coefficients: An evaluation of robustness MP3 encoded music," in *Proc. of the 7th International Society for Music Information Retrieval Conference (ISMIR 06)*, (Victoria, Canada), 2006.

[51] K. Karplus and A. Strong, "Digital synthesis of plucked-string and drum timbres," *Computer Music Journal*, vol. 7, no. 2, pp. 43–55, 1983.

[52] V. Välimäki, J. Huopaniemi, and Z. Jánosy, "Physical modeling of plucked-string instruments with application to real-time sound synthesis," *Journal of Audio Engineering Society*, vol. 44, no. 5, pp. 331–353, 1996.

[53] C. Erkut, V. Välimäki, M. Karjalainen, and M. Laurson, "Extraction of physical and expressive parameters for model-based sound synthesis of the classical guitar," in *Proc. of 108th AES International Convention*, (Paris, France), 2000.

[54] R. V. Migneco and Y. E. Kim, "Excitation modeling and synthesis for plucked guitar tones," in *Proc. of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, (New Paltz, NY), pp. 193–196, Oct. 2011.

[55] D. A. Jaffe and J. O. Smith, "Extensions of the Karplus-Strong plucked string algorithm," *Computer Music Journal*, vol. 7, no. 2, pp. 56–69, 1983.

[56] J. O. Smith and S. A. V. Duyne, "Commuted piano synthesis," in *Proc. of International Computer Music Conference (ICMC-95)*, (Banff, Canada), pp. 319–326, 1995.

[57] N. Lindroos, H. Penttinen, and V. Välimäki, "Parametric electric guitar synthesis," *Computer Music Journal*, vol. 35, no. 3, pp. 18–27, 2011.

[58] P. M. Morse, *Vibration and sound*. American Institute of Physics for the Acoustical Society of America, 1981.

[59] Fender Musical Instruments Corporation, "American standard stratocaster."

[60] D. Halliday, R. Resnick, and J. Walker, *Fundamentals of Physics*. Wiley, sixth ed., 2000.

[61] M. Karjalainen, V. Välimäki, and T. Tolonen, "Plucked-string models: From the Karplus-Strong algorithm to digital waveguides and beyond," *Computer Music Journal*, vol. 22, no. 3, pp. 17–32, 1998.

[62] N. H. Fletcher and T. D. Rossing, *The Physics of Musical Instruments*. Springer-Verlag, second ed., 1998.

[63] B. Bank and V. Välimäki, "Robust loss filter design for digital waveguide synthesis of string tones," *IEEE Signal Processing Letters*, vol. 10, no. 1, pp. 18–20, 2003.

[64] G. Evangelista and M. Raspaud, "Simplified guitar bridge model for the displacement wave representation in digital waveguides," in *Proc. of International Conference on Digital Audio Effects (DAFx-09)*, (Como, Italy), Sept. 2009.

[65] H.-M. Lehtonen, J. Rauhala, and V. Välimäki, "Sparse multi-stage loss filter design for waveguide piano synthesis," in *Proc. of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, (New Paltz, NY), pp. 331–334, Oct. 2005.

[66] S. Cho, H. Han, J. Kim, U. Chong, and S. Cho, "Modification of the loop filter design for a plucked string instrument," *Journal of Acoustical Society of America*, vol. 131, no. 2, pp. EL126–EL132, 2012.

[67] B. Bank and V. Välimäki, "Passive admittance matrix modeling for guitar synthesis," in *Proc. of International Conference on Digital Audio Effects (DAFx-10)*, (Graz, Austria), pp. 3–9, Sept. 2010.

[68] M. van Walstijn, "Parametric FIR design of propagation loss filters in digital waveguide string models," *IEEE Signal Processing Letters*, vol. 17, no. 9, pp. 795–798, 2010.

[69] H. Fletcher, E. D. Blackham, and R. Stratton, "Quality of piano tones," *Journal of Acoustical Society of America*, vol. 13, no. 1, pp. 749–761, 1962.

[70] S. A. V. Duyne and J. O. Smith, "A simplified approach to modeling dispersion caused by stiffness in strings and plates," in *Proc. of International Computer Music Conference (ICMC-94)*, (Aarhus, Denmark), pp. 407–410, September 1994.

[71] D. Rocchesso and F. Scalcon, "Bandwidth of perceived inharmonicity for physical modeling of dispersive strings," *IEEE Transactions on Speech and Audio Processing*, vol. 7, no. 5, pp. 407–414, 1999.

[72] J. Rauhala and V. Välimäki, "Tunable dispersion filter design for piano synthesis," *IEEE Signal Processing Letters*, vol. 13, no. 5, pp. 253–256, 2006.

[73] J. S. Abel and J. O. Smith, "Robust design of very high-order allpass dispersion filters," in *Proc. of International Conference on Digital Audio Effects (DAFx-06)*, (Montreal, Canada), pp. 13–18, September 2006.

[74] J. Rauhala, H.-M. Lehtonen, and V. Välimäki, "Fast automatic inharmonicity estimation algorithm," *Journal of Acoustical Society of America*, vol. 121, no. 5, pp. EL184–EL189, 2007.

[75] S. Haykin, *Adaptive Filter Theory*. Prentice Hall, 4th ed., 2004.

[76] T. Kailath and A. H. Sayed, *Linear Estimation*. Prentice Hall, 1st ed., 2000.

[77] G. Fant, J. Liljencrants, and Q. Lin, "A four-parameter model of glottal flow," *STL-QPSR*, vol. 4, pp. 1–13, 1985.

[78] H. Li, R. Scaife, and D. O'Brien, "LF model based glottal source parameter estimation by extended Kalman filtering," in *Proc. of IET Irish Signals and Systems Conference (ISSC 2011)*, (Dublin, Ireland), June 2010.

[79] G. Cuzzucoli and V. Lombardo, "A physical model of the classical guitar, including the players touch," *Computer Music Journal*, vol. 23, no. 2, pp. 52–69, 1999.