HELSINKI UNIVERSITY OF TECHNOLOGY
Department of Electrical and Communications Engineering
Laboratory of Acoustics and Audio Signal Processing

Jyri Pakarinen

Spatially Distributed Computational Modeling of a Nonlinear Vibrating String

Master's Thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Technology.

Espoo, June 2, 2004

Supervisor: Professor Vesa Välimäki

HELSINKI UNIVERSITY OF TECHNOLOGY

ABSTRACT OF THE MASTER'S THESIS

Author:	Jyri Pakarinen
Name of the thesis:	Spatially distributed computational modeling of a nonlinear vibrating string
Date:	June 2, 2004 **Number of pages:** 59
Department:	Electrical and Communications Engineering
Professorship:	S-89 Acoustics and Audio Signal Processing
Supervisor:	Prof. Vesa Välimäki
Instructor:	Prof. Vesa Välimäki

Nonlinearities in string instruments are responsible for several interesting acoustical features, resulting in characteristic and easily recognizable tones. For this reason, modern synthesis models have to be capable of modeling this nonlinear behavior, when high quality results are desired. This thesis presents two novel physical modeling algorithms for simulating the tension modulation nonlinearity in plucked strings in a spatially distributed manner. The first method uses fractional delay filters within a digital waveguide structure, allowing the length of the string to be modulated during run time. The second method uses a nonlinear finite difference approach, where the string state is approximated between sampling instants also using fractional delay filters, thus allowing run-time modulation of the temporal sampling location. The magnitude of the tension modulation is evaluated from the elongation of the string at every time step in both cases. Simulation results of the two models are presented and compared. Real-time sound synthesis of the kantele, a traditional Finnish plucked-string instrument with strong effect of tension modulation, has been implemented using the nonlinear digital waveguide algorithm.

Keywords: sound synthesis, tension modulation, digital waveguide, finite difference time-domain modeling, computer music.

TEKNILLINEN KORKEAKOULU DIPLOMITYÖN TIIVISTELMÄ

Tekijä:	Jyri Pakarinen
Työn nimi:	Epälineaarisen värähtelevän kielen paikkajakautunut laskennallinen mallinnus
Päivämäärä:	2. kesäkuuta 2004 **Sivuja:** 59
Osasto:	Sähkö- ja tietoliikennetekniikka
Professuuri:	S-89 Akustiikka ja äänenkäsittelytekniikka
Työn valvoja:	Prof. Vesa Välimäki
Työn ohjaaja:	Prof. Vesa Välimäki

Värähtelevän kielen epälineaarinen käyttäytyminen saa monissa kielisoittimissa aikaan soittimelle luonteenomaisen ja helposti tunnistettavan äänen. Laadukkaan kielisoitinsynteesin vuoksi onkin tärkeää, että nykyaikaiset äänisynteesimenetelmät ottavat huomioon myös kielten epälineaarisuudet. Tässä diplomityössä esitellään kaksi uutta synteesimenetelmää, jotka fysikaalisen mallinnuksen avulla simuloivat epälineaarisia näpättyjä kieliä paikkajakautuneesti, keskittyen jännitysmodulaation tuottamiin epälineaarisuuksiin. Toinen menetelmistä käyttää hajautettuja murtoviivesuotimia digitaalisen aaltojohtomallin viivesilmukan pituuden ajonaikaisessa virittämisessä, kun taas toinen hyödyntää murtoviivesuotimia äärelliseen erotukseen pohjautuvan mallin aikaresoluution muuttamisessa ajon aikana. Jännitysmodulaation suuruus arvioidaan kummankin mallin tapauksessa jokaisella aika-askeleella kielen pidentymästä. Molempien mallien simulaatiotulokset esitellään ja niitä verrataan toisiinsa sekä myös mitattuihin arvoihin. Epälineaarisen aaltojohtomallin avulla on toteutettu reaaliaikainen kantelemalli.

Avainsanat: Äänisynteesi, jännitysmodulaatio, digitaaliset aaltojohdot, äärellisen erotuksen aika-alueen mallinnus, tietokonemusiikki

Acknowledgements

This Master's thesis has been carried out in the Laboratory of Acoustics and Audio Signal Processing at Helsinki University of Technology. The research was funded by the European Union 5th Framework Future and Emerging Technologies ALMA-project (IST-2001-33059).

Firstly, I want to thank my supervisor Professor Vesa Välimäki for his guidance and support. His comments on both scientific and grammatical issues have improved this thesis considerably. I would also like to thank Professor Matti Karjalainen and Doctor Cumhur Erkut for their advice, ideas, and several interesting discussions.

Finally, I would like to thank my loving fiancée Ms. Annika Peurala for being there and standing by me.

Otaniemi, June 2, 2004

Jyri Pakarinen

Contents

List of Symbols

α	[rad]	Angle between the string and the soundboard at the termination
α_{TMDF}		Scaling coefficient for the missing harmonics in the nonlinear FDTD string
γ		Spectral amplification function
ϵ	$[kg/m]$	Linear mass density
θ	[rad]	Spatial frequency
λ_n	$[m]$	Wavelength of the nth wave component
τ	[s]	Time constant for frequency-independent decay
a		Allpass filter coefficient
a_m		Finite difference loss parameter
a_p		Leaky integrator parameter
c	$[m/s]$	Transversal wave propagation velocity
c_l	$[m/s]$	Longitudinal wave propagation velocity
d		Fractional delay value
d_1	$[Kg/(ms)]$	Damping coefficient simulating frequency-independent losses
d_2	$[Kgm^2/s^2]$	Damping coefficient simulating frequency-dependent losses
d_3	$[Nm^2]$	Stiffness coefficient
$d_{partial}$		Delay value for a single basic element
e		Base for natural logarithms ($e \approx 2.7183$)
f	$[Hz]$	Frequency
f_1	$[Hz]$	Fundamental frequency
$f_{in}(n)$		Force exciting the kantele string model
$f_{out}(n)$		Force excerted by the kantele string to the string termination point
f_s	[Hz]	Temporal sampling frequency
$f_x(n)$		Force excerted by the tension modulation to the termination point
$f_y(n)$		Force excerted by the horizontal string vibration to the termination point
$f_z(n)$		Force excerted by the vertical string vibration to the termination point
$f_{zy}(n)$		Force excerted by the vertical string vibration to the horizontal vibration
g		Loss coefficient in the DWG model simulating frequency-independent losses

g_m		Finite difference loss parameter
g_m^+		FDTD waveguide scattering coefficient
g_m^-		FDTD waveguide scattering coefficient
g_p		Leaky integrator gain parameter
$l_{dev}(t)$	$[m]$	Elongation of the string
l_{nom}	$[m]$	Nominal length of the string
m		Spatial sample index
n		Temporal sample index
r		Tuning coefficient a.k.a. Courant number
r_l		Left reflection coefficient of a lossy finite difference termination
r_r		Right reflection coefficient of a lossy finite difference termination
r_s	$[m]$	Radius of the string
$s(n)$		Slope of the string displacement at time instant n
$s_l(n,k)$		Slope of the wave component propagating left at time instant n and location k
$s_r(n,k)$		Slope of the wave component propagating right at time instant n and location k
t	$[s]$	Time variable
$v(n)$		Transversal wave velocity at time instant n
$v_y(n)$		Horizontal wave velocity at time instant n
$v_z(n)$		Vertical wave velocity at time instant n
w		Torsional component of the string vibration
x, y, z	$[m]$	Spatial coordinates of a string segment
y_l		Displacement of the wave component propagating left
y_r		Displacement of the wave component propagating right
$y(n)$		String displacement at time instant n
A	$[m^2]$	Cross-sectional area of a string
C		Scaling coefficient for the direct coupling between TM and instrument body
$D_{tot}(n)$		Total delay variation of a nonlinear DWG at time instant n
$F(n)$		Exciting force at time instant n
E	$[N/m^2]$	Young's modulus
$H(z)$		Loss filter
K	$[N]$	String tension
K_0	$[N]$	Nominal string tension
K_x	$[N]$	Longitudinal component of string tension
K_z	$[N]$	Vertical component of string tension
L	[s]	Delay line length
$L_{dev}(n)$		Approximated elongation of a DWG string at time instant n
L_{nom}		Nominal length of a DWG string

$\hat{L}_{nom}$		Rounded nominal length of a DWG string
$O(\cdot)$		Order of the approximation error
T	[s]	Length of the temporal sampling interval
X	[m]	Length of the spatial sampling interval
Z_1, Z_2		Mechanical impedances of string segments
Z_c		Mechanical impedance from the vertical to horizontal string vibration
$Z_{ybridge}$		Mechanical impedance in the horizontal direction between the string and the termination
$Z_{zbridge}$		Mechanical impedance in the vertical direction between the string and the termination

Abbreviations

1D	One-dimensional
3D	Three-dimensional
DSP	Digital Signal Processing
DWG	Digital Waveguide
EVE	Experimental Virtual Environment
FD	Fractional Delay
FDM	Finite Difference Method
FDTD	Finite Difference Time Domain
FIR	Finite Impulse Response
FM	Frequency Modulation
IIR	Infinite Impulse Response
KS	Karplus-Strong
LTI	Linear and Time-Invariant
MFLOPS	Millions of Floating-Point Operations Per Second
NDWG	Nonlinear Digital Waveguide
NFDTD	Nonlinear Finite Difference Time Domain
TM	Tension Modulation
TMDF	Tension Modulation Driving Force
WDF	Wave Digital Filter

Chapter 1

Introduction

This thesis deals with physical modeling of nonlinearly vibrating strings from the sound synthesis point of view. The purpose of sound synthesis in general is to electronically produce both realistic and non-realistic sounds for the needs of electronic musicians, computer games, the movie industry, and virtual reality applications. The task of this thesis is to provide computational algorithms which simulate the behavior of nonlinear strings in a physically meaningful way. These algorithms can then be used to produce synthesized string instrument sounds with controllable physical parameters using a computer or a signal processor. The novelty of this thesis lies in two new computational models: a spatially distributed nonlinear digital waveguide string model, recently published by the author [42], and a spatially distributed nonlinear finite difference string model.

1.1 Sound Synthesis Through Non-physical Methods

Until approximately the 1990s, sound synthesis was mainly implemented using techniques which focused on imitating the sound itself, rather than the system that produces it. The simplest of these techniques, the sampling synthesis, merely uses pre-recorded samples of the sound and plays them back [53]. A slightly more advanced method, the wavetable synthesis, uses combinations of short samples of different waveforms and concatenates or sums them up in order to resemble the desired wave [53]. These sample-based methods, although widely used in the commercial market and capable of producing very high-quality sounds, require quite a lot of memory and lack the flexibility of adjusting the properties of the sound in an intuitive way. A third sample-based method, granular synthesis, consideres sound signals as consisting of "sound atoms" or grains, and combines them pitch synchronously or asynchronously for generating the synthesized sound [53].

Another computationally simple sound synthesis technique, frequency modulation (FM), became popular in the synth-pop of the 1980s. Originally introduced by Chowning in 1973

[14], FM synthesis in its basic form consists of a sine oscillator which is used to control the pitch of another sine oscillator, thus producing a rich spectral content. This computationally straightforward technique requires virtually no memory, and is capable of synthesizing highly inharmonic sounds such as bells, cymbals, etc. Adjusting the FM synthesis control parameters, however, results in even more ambiguous changes in the sound than with the previously discussed models.

Spectral synthesis methods, as their name implies, try to simulate the spectral properties of the target sound as they are perceived by the listener. One of these methods, additive synthesis, is based on the idea that every sound can be seen as a superposition of an infinite number of sine oscillators, each having different amplitude, frequency, and phase. Additive synthesis uses a high number of these oscillators and sums their outputs to simulate the target sound. Since each of the oscillators consume nonzero process time in reality, the sound quality of additional synthesis is directly proportional to the computational complexity of the system.

Also belonging to the family of spectral synthesis methods, source-filter synthesis uses spectrally rich source signals together with time-varying filters to produce the synthesized sound. This method, also known as subtractive synthesis, bears resemblance to the human voice production mechanism, where the spectrally complex glottal signal is filtered by the vocal tract. Depending on the case, this synthesis technique can be sometimes also seen as a physical modeling technique. There are also several other useful and efficient techniques for sound synthesis, but the list of all conventional synthesis methods is simply too numerous to be discussed here. More thorough treatment of different sound synthesis techniques in general can be found e.g. in [62] or [53].

1.2 Sound Synthesis Through Physical Modeling

A common disadvantage in the synthesis techniques in the preceding discussion is that the synthesis control parameters, i.e. the parameters that change the tonal qualities of the synthesized sound, are rather unintuitive and proper tuning of them is not based on real-life physics. This results in synthesizers, which are hard to control and lack the expressiveness of real musical instruments. To answer this call for intuitively-controlled sound synthesis algorithms, sound synthesis through physical modeling has emerged.

Physical modeling is based on the idea that a real-life system can be simulated using mathematics, and the behavior of this computational model closely mimics the behavior of the simulation target. In the sound synthesis context, this means that if we want to model, say a drum, we will start by studying the vibrational behavior of a real drum, formulate the related mathematical expressions, and program them on a computer or a signal processor. After this, we can excite our physical model e.g. by hitting a pressure sensitive tablet connected to the computer, and hear the corresponding sound via the computer's loudspeakers. Since the com-

putational model is based on physics, we can control various features of the drum model, e.g. the material and dimensions of the membrane and the drum body, sound propagation velocity inside the drum, etc.

It is also possible to construct physical models that would not be possible to implement in real life, such as a drum with time-varying dimensions, or to connect different models in an unorthodox way (e.g. to make a guitar string which is excited with a clarinet reed.). The control of the physical models can be made even more intuitive if the model parameters are obtained using electric sensors, such as accelerometers, wind controllers, or pressure sensors. Then a musician could play this virtual instrument in a similar way that real musical instruments are played, but would have the freedom of changing the model parameters and structure, if needed.

The different physical modeling techniques can roughly be divided into six categories [68]:

1. Source-filter models
2. Modal synthesis
3. Mass-spring networks
4. Wave digital filters
5. Digital waveguide modeling
6. Finite difference methods

First of these, the source-filter models, were already briefly discussed in the previous section. Source-filter models of the human voice-production mechanism can be considered as physical models, if both the source and the filter have physically meaningful presentations (e.g. a classic linear-prediction speech production model is not a physical model since the glottal source is typically seen as a mere impulse generator). Singing synthesis through source-filter models has been studied e.g. in [54].

Modal synthesis relies on the fact that any sound-producing object can be represented as a collection of vibrating substructures characterized by modal data [2]. The modal data associated with a specific substructure consists of the frequencies and the damping coefficients of the structure's resonant modes, and of a set of coordinates representing the shapes of the vibration mode. Generally, the modal data from different structures has been obtained via vibrational measurements. These modal data then respond to external excitations and to interaction forces or air flows.

When a general external force is applied to a given point on a structure, it excites the structure's modes. Every mode responds to this excitation independently of the other modes, but the responding force and the resulting velocity is again the superposition of the forces and the velocities of the individual modes. Therefore, in modal synthesis, the internal construction of a complex vibrating structure can be seen as a set of parallel independent resonators and can

be implemented as a set of parallel digital filters. The exciting motion to the substructure is characterized by coordinates, which are bound to different filters by modal excitation matrices. These matrices define the relationship between the place of excitation and the resulting modal behavior. Basically, this means that excitation to different parts of the substructure excites different modes and produces different sounds; this simulates closely the behavior of real vibrating structures.

A recent technique related to modal synthesis is the *functional transformation method* [64], [63], where the modal data can be derived directly from the underlying wave equations via Laplace- and Sturm-Liouville transformations. This is a considerable improvement, since it removes the need for vibrational measurements and enables also the modeling of non-realizable structures. After the algorithms for different vibrating objects have been derived, they can efficiently be implemented in real time using digital filters.

Mass-spring networks consider physical structures as lattices of point masses interconnected by springs. In a way, this can be seen as an "atomic view" of vibrating systems, since the structures are divided into sets of discrete mass-spring oscillators. Although the principles of mass-spring networks are relatively straightforward to understand, modeling of more sophisticated phenomena, e.g. frequency-dependent losses, stiffness, or nonlinearities is not so simple. Also, since the number of modes in the synthesized sound is directly proportional to the number of mass-spring oscillators in the model, high quality sound requires quite a lot of computational power. Computer simulation of mass-spring networks has been discussed in [22].

Wave digital filters (WDFs) have their background in electrical engineering, where they were used in discretizing analog filters into digital ones already in the 1970s. This technique, first introduced by Fettweis [20], can as well be used in the modeling of vibrating systems, if the mechanical system is first converted into an equivalent electric one using certain well-known analogies. In the context of musical instruments this would mean that the system to be modeled, say a piano hammer, must first be formulated as a mechanical system (e.g. as a mass representing the hammer core, connected to a spring representing the compressible felt). This mechanical structure can then be represented as an electrical circuit (a parallel capacitor-inductor circuit, to be exact), and the circuit can be discretized using e.g. the *bilinear transformation*. The discretization effectively re-defines the voltages and currents as incoming and outgoing wave components. WDFs have usually been used in the modeling of lumped systems [5], [47], although the modeling of spatially distributed systems is also possible using multidimensional WDFs [7].

Digital waveguides and finite difference methods are extensively used in the physical modeling of musical instruments, and they will be more thoroughly discussed in Chapters 3 and 4, respectively.

1.3 Contents of This Thesis

This thesis is structured as follows: we will begin by studying the elementary physics of vibrating strings in Chapter 2, where the foundation for understanding the latter part of this thesis is laid. The next chapter, Chapter 3, will start by introducing the reader to *digital waveguide* (DWG) modeling, and continue to formulate a spatially distributed digital waveguide string model with tension modulation nonlinearity. Stability of the nonlinear DWG algorithm is also discussed. The final section of Chapter 3 will give an example of how a physical model of a *kantele*, a Finnish plucked-string instrument, can be implemented using the nonlinear DWG model, and also present simulation results with comparisons to measured data.

Chapter 4 will begin by explaining the basic concepts of the *finite difference method* (FDM), and then proceed to introducing a nonlinear finite difference string model. Stability issues, simulation results, and comparisons to real-world cases are also presented in this chapter, as well as a comparison between the two nonlinear models. Chapter 5 will finally present conclusions and point the direction to possible future studies.

Chapter 2

Physical Analysis of a Vibrating String

Since the purpose of physics-based modeling is to simulate the physical phenomena in the system of interest, it is of paramount importance to be familiar with the behavior of the real-world case before modeling can take place. This chapter will study some physical properties of vibrating strings, the main focus being on those, which contribute the most to the resulting sound.

2.1 Linear String

This section covers some of the underlying physics of a simplified vibrating string. The phenomena discussed here has been presented in various textbooks on physics and acoustics (e.g. [21], [17]), and it forms a basis for the comprehension of the upcoming chapters.

In mathematics, a function $f(\cdot)$ is said to be linear if and only if $f(\alpha x + \beta y) = \alpha f(x) + \beta f(y)$. This could be translated to plain English by stating that a system is considered to be linear if its response is directly proportional to its excitation and if it treats combined excitation signals as if they were independent of each other. As an example of the former, in a linear string, if a plucking with some force F_0 would result to a vibrational amplitude of A_0, a plucking with a twice as high force $2F_0$, would result to twice as high vibrational amplitude $2A_0$. Another useful concept is the time invariability, which in plain terms means that a time invariable system operates in the exact same way at all times. If a system meets both of these criteria, it is said to be *linear and time-invariant* (LTI). LTI systems are of great importance because they can be commuted with each other, that is, two LTI systems can be merged into one single equivalent system.

Although no real strings behave exactly like this, we can often simplify things by assuming LTI, especially, if only modest vibrational amplitudes are considered. In this section we'll discuss this simplified case in order to derive expressions for string movement.

2.1.1 Vibrational polarizations

The movement of a vibrating string can be divided in four polarizations. This means that each infinitely small string segment has four components of motion. The different polarizations are illustrated in Figure 2.1. The first two of them, called transversal polarizations, take place in the direction perpendicular to the string. These polarizations are named *horizontal* and *vertical*, respectively. The next polarization is the movement in the direction of the string, i.e. the compressional waves within the string medium. This polarization is called *longitudinal*. The remaining polarization is called *torsional*, and it describes the rotational motion of the string segment around its axis. The same naming policy for the polarizations is used throughout this thesis. In plucked string instruments, the sound is mainly generated by string's transversal motion, although the longitudinal vibration often adds some interesting effects. The torsional polarization is important with string-bow interactions (e.g. with a violin or a cello), but has little effect in plucked strings and is therefore not discussed in this thesis. The coordinate axes are assigned for different polarizations as follows: the longitudinal coordinate is denoted by x, the horizontal coordinate by y, and the vertical coordinate by z.

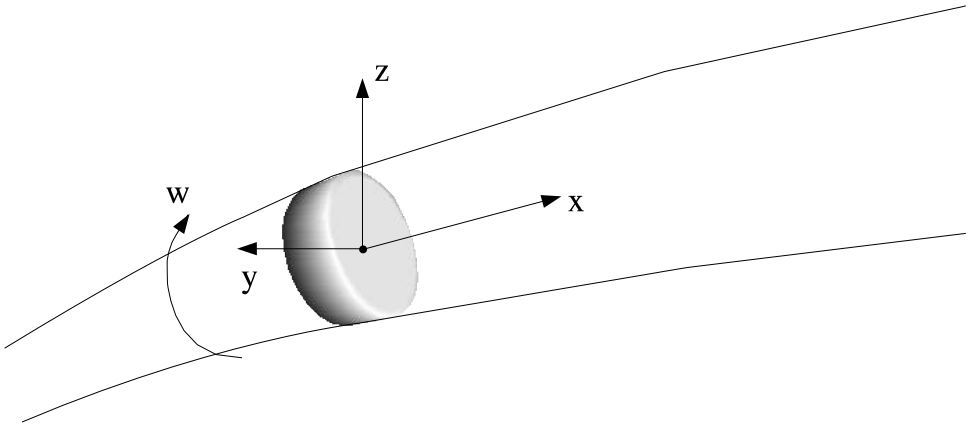

Figure 2.1: Each string segment has four components of motion, a.k.a four polarizations. These are called longitudinal, horizontal, vertical, and torsional, and noted in this figure by x,y,z, and w, respectively.

2.1.2 Ideal string

Let us consider for a moment a homogeneous string, which is completely flexible, linear and lossless (i.e. the string's total energy remains constant). Such a string is called an ideal string. If we also consider the string moving only in one transversal polarization (e.g. horizontal), the motion of an ideal string can be characterized by the well-known one-dimensional (1D) wave equation:

$$y_{tt}(t,x) = c^2 y_{xx}(t,x). \tag{2.1}$$

Here, $y_{tt}(t,x)$ denotes the second-order partial derivative of the string displacement in the horizontal axis with respect to the time variable t, $y_{xx}(t,x)$ denotes the second-order partial derivative of the horizontal displacement with respect to the longitudinal coordinate x, and c denotes the transversal wave propagation velocity within the string medium. The first term of Eq. 2.1, $y_{tt}(t,x)$, can be seen as the acceleration of a string segment at time instant t and position x, and $y_{xx}(t,x)$ can be seen as the curvature of the string at the same point. Also, the transversal wave propagation velocity can be written as

$$c = \sqrt{\frac{K}{\epsilon}}, \tag{2.2}$$

where K is the string tension and ϵ is the linear mass density of the string. By rewriting Eq. 2.1 as

$$Ky_{xx}(t,x) = \epsilon y_{tt}(t,x), \tag{2.3}$$

we can see that the 1D wave equation can be interpreted as an expression of Newton's second law *"force = mass × acceleration"* on a microscopic scale [21], [17].

The 1D wave equation for longitudinal vibrations is similar to Eq. 2.1, but with the transversal wave propagation velocity c replaced with the longitudinal wave propagation velocity

$$c_l = \sqrt{\frac{EA}{\epsilon}}, \tag{2.4}$$

where E is Young's modulus and A is the cross-sectional area of the string.

2.1.3 Traveling-wave solution

One solution to the 1D wave equation (Eq. 2.1) can be seen as a superposition of two waveforms traveling in opposite directions along the string. Figure 2.2 clarifies this interpretation. This solution, commonly known as the *traveling-wave solution*, or as *d'Alembert's solution*, was first published by d'Alembert in 1747. This solution can be presented in the mathematical language as [56]

$$y(t,x) = y_r(t - x/c) + y_l(t + x/c), \tag{2.5}$$

where y_r and y_l denote the wave components proceeding right and left, respectively.

In an ideal string with fixed terminations, a traveling wave encounters a phase-inverting reflection when reaching the endpoint. Also, the total string motion is a linear superposition of all wave components traveling in the string. Eventually, this results in a vibrational movement consisting only of such standing waves whose wavelengths satisfy

$$\lambda_n = 2\frac{x_L - x_0}{n}, n = 1, 2, 3, ... \tag{2.6}$$

where λ_n is the wavelength of the nth wave component and $x_L - x_0$ denotes the string's length. This restriction can be understood for example by considering the fact that the string is fixed at

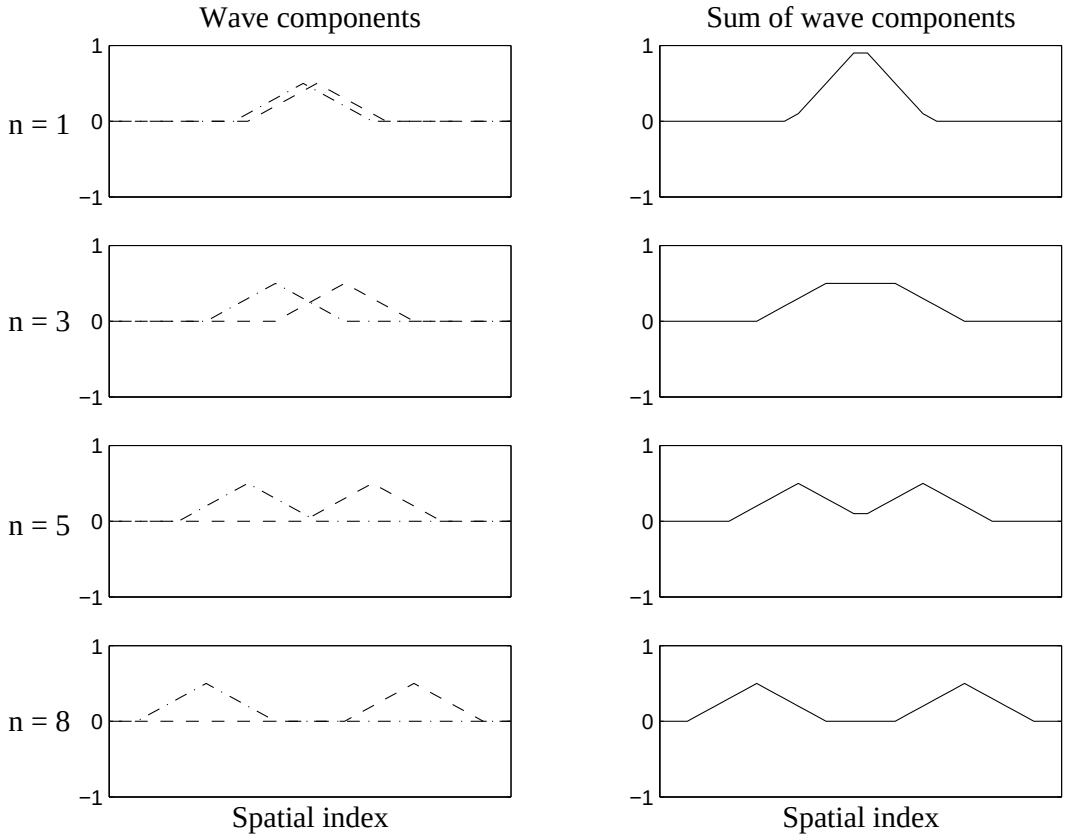

Figure 2.2: [55] A vibrating string can be seen as consisting of two wave components, traveling at opposite directions. In this illustration, the left column shows the wave components at different time steps n. The dashed line denotes the wave component traveling right, while the dash-dotted line denotes the wave component traveling left. The sum of the wave components (a.k.a. the string displacement) at the same time instants is illustrated in the right column. The string is plucked at three points simultaneously, thus yielding a triangular initial state.

its ends, i.e. the string displacement must be zero at $x = 0$ and $x = L$. Since the standing waves are sinusoidal, they will have zero crossings at mid-wavelength and at wavelength. Therefore they will also have to satisfy Eq. 2.6.

The different values of n in Eq. 2.6 are called the vibrational *modes* of the string, and the total string vibration is formed by the superposition of these modes. Since the relationship between frequency and wavelength is $f = 1/(2\pi\lambda)$, each mode produces a different vibrational frequency which is an integer multiple of the fundamental frequency $f_1 = 1/(2\pi\lambda_1)$. These frequencies are located at harmonic intervals in the frequency scale, and are therefore called *harmonics*. Note that the first harmonic f_1 is also considered the fundamental frequency, since it is the lowest vibrational component in the system. The other harmonics have integer relation to the fundamental frequency, so that $f_k = kf_1, k = 2, 3, 4, ...$ It is worth noting that there

is some deviation in the numbering of the harmonic indices in the literature, since in speech processing the fundamental frequency is often denoted f_0. The harmonics are often referred also as *partials*, because they do not necessarily always have harmonic intervals. We will study this case in more detail when discussing stiff strings in Section 2.1.5.

If an ideal string is plucked at $x_p = (x_L - x_0)/n_p$, it is obvious that the resulting waveform cannot have a nodal point at x_p. Therefore also, every n_pth mode will be missing, and also every n_pth harmonic in the resulting sound. Figure 2.3 illustrates the first vibrational modes of a string, when it is plucked at different locations. These wave components which are absent from the vibrational spectrum in the linear case are called *missing harmonics*, and they will be discussed more thoroughly in Section 2.2.3.

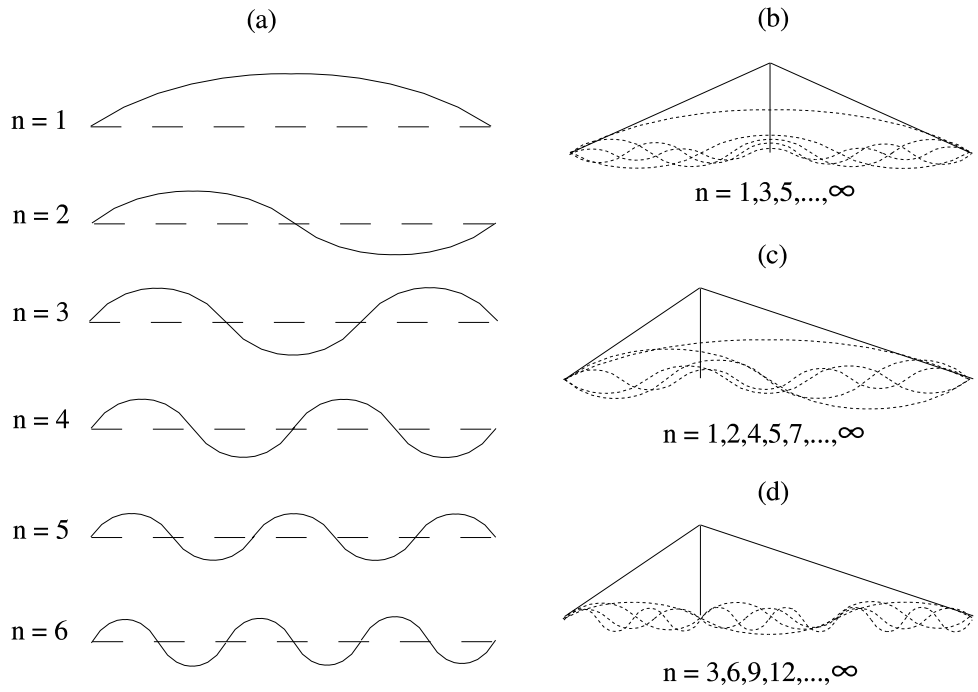

Figure 2.3: In (a) the first six vibrational modes of a string are presented. The total vibrational behavior of the string will be the superposition of all its modes. In (b), a string plucked at the middle and its first modes are shown. Due to plucking location, the vibration consists only of odd partials. Case (c) shows the string plucked at $1/3$rd of its length. Again, since the vibration cannot have a nodal point at the plucking location, every 3rd mode is missing. The modes that are missing in (c) are shown in (d).

2.1.4 Lossy string

Obviously, every real string vibration decays with time. This results mainly from three damping mechanisms [21]: (1) air damping, (2) internal damping in the string medium, and (3) mechanical energy transfer through string terminations. The first two mechanisms are caused by viscoelastic and thermodynamic losses and are therefore irreversible. Mechanical energy transfer is a bit more complex phenomenon since it is a reversible process i.e. the energy can move back and forth via the terminations.

Air damping takes place when the viscous flow around the vibrating string retards its motion. Since the radii of strings are usually small, direct sound radiation can be neglected as a damping term. Internal damping is caused by the elastic behavior of the string, and it is defined by the string material only, while energy loss through the supports is characterized by the mechanical admittances of the terminations.

Each of the damping mechanisms introduces an exponential decay to string oscillations and has an individual frequency-dependent time constant. The time constant for the total decay behavior of the string can be obtained by adding the reciprocals of the individual time coefficients. As stated in [21], for thin metal strings, the losses consist mainly of air damping, and the decay time for the upper harmonics is proportional to $1/f$. For instruments with gut or nylon strings, however, internal losses become most dominant, and the decay time for the upper harmonics varies as $1/f^2$. Therefore, metal strings often have a considerably brighter tone than nylon or gut strings.

The air damping mechanism is proportional to the transversal velocity component of the string segment, and can therefore be inserted to the 1D wave equation 2.3 as a term containing a first-order temporal derivative. Frequency-dependent losses, such as the internal friction of the string, can be approximated by inserting a loss term containing a third-order temporal derivative into the same equation, ending up with the lossy 1D wave equation [26], [11]:

$$Ky_{xx}(t,x) = \epsilon y_{tt}(t,x) + d_1 y_t(t,x) - d_2 y_{xxx}(t,x). \tag{2.7}$$

Here, d_1 and d_2 are coefficients that simulate the frequency-independent and frequency-dependent damping, respectively. The third-order spatial derivative term could be replaced with a term containing mixed spatial and temporal derivatives as suggested in [63], but it would not likely yield more accurate results, since both of them are merely harsh approximations of the real-world case. A more sophisticated damping model can be found in [13] and [38].

2.1.5 Stiff string

In real strings, the restoring force consists not only of the string tension, but also of the stiffness of the string. In flexible strings with small diameter (e.g. guitar), the effect of stiffness can usually be neglected, but with larger strings (e.g. piano strings) it contributes substantially to

the resulting sound. We can obtain the 1D wave equation for a stiff, lossy string by inserting an additional term to Eq. 2.7, yielding

$$Ky_{xx}(t,x) = \epsilon y_{tt}(t,x) + d_1 y_x(t,x) - d_2 y_{txx}(t,x) + d_3 y_{xxxx}(t,x), \tag{2.8}$$

where $y_{xxxx}(t,x)$ is the fourth-order spatial derivative of the string displacement, and [21]

$$d_3 = \frac{E\pi r_s^4}{4},$$

where r_s stands for the radius of the string. In stiff strings, the wave components with different frequencies travel with different velocities. This phenomenon is called *dispersion*, and it affects the resulting spectrum so that the partials are no longer located at exactly harmonic intervals, but are stretched along the frequency axis. This inharmonicity is taken into account for example when tuning a piano, where the octaves are tuned to greater ratios than 2:1 to minimize the beating between different strings. Since this thesis focuses on the sound synthesis of strings with relatively high elasticity and small diameters, the effects of stiffness are not covered thoroughly here. More in-depth study about this topic can be found for example in [51].

2.2 Effects Due to Nonlinearities

The vibration of real strings can be considered linear in the most coarse approximations only. More realistic string models require abandoning the linearity assumption, thus also introducing more complex formulations for string behavior. This section will study some of the nonlinearities in real strings.

2.2.1 Tension modulation

When a real string is displaced, its length, and therefore also its tension is increased. When the string returns closer to its equilibrium state, its length and tension are decreased. This mechanism, where the tension is varied due to transversal vibrations, is called *tension modulation*, and abbreviated TM. In the following, K_0 is used to denote the *nominal tension* of the string, i.e. the string tension at rest.

It is easy to see that the frequency of the TM is twice the frequency of the transversal vibration, since the transversal equilibrium produces minimum tension, and both extreme diplacements produce maximum tension. In many musical instruments the string termination is rigid in the longitudinal direction, so that TM cannot be effectively coupled to the instrument body, and therefore cannot be heard directly. In instruments where this is not the case, partials created by TM can be found at twice the frequencies of the transversal modes, and they are called *phantom partials*. The generation of these partials is discussed in more detail in [15], and in [4]. Other effects of TM are discussed in the following two subsections.

2.2.2 Initial pitch glide

Due to decaying vibration, the string displacement is clearly greater just after the pluck than at the end of the vibration. Therefore also the amplitude envelope of TM has the same form as the transversal vibration itself. Also, the average value of the tension has a greater value in the beginning of the vibrational movement, and it gradually decreases towards the nominal tension level K_0.

As can be seen from Eq. 2.2, this increase in tension results in the increase of wave velocities, which in turn leads to the increase in frequency. Thus, the frequency of a plucked string glides from an initial value to the steady-state value (the frequency in the linear case). This is called *initial pitch glide*, an effect which is most apparent with elastic strings with large vibrational amplitudes and relatively low nominal string tension (such as electric guitar or kantele strings).

2.2.3 Generation of missing harmonics

Tension modulation is also responsible for another acoustic feature, *generation of missing harmonics*. This means, as its name implies, that the harmonics which should be missing from the vibration can be found from the spectrum. Typically, they begin with a gradual increase near zero amplitude until they reach their peak value, and then decay off like all other harmonics. In the following, this phenomenon is discussed, while trying not to go too deep into the mathematics. A more in-depth study can be found from a paper by Legge and Fletcher [39].

Let us assume that the string tension K is small enough, so that the transversal wave velocities are considerably smaller than the longitudinal wave velocity (in reality, the longitudinal wave velocity is typically only 5-20 times higher than the transversal one, see Eqs. 2.2 and 2.4). This means that the tension can be taken as uniform along the string. Although the TM takes place in the longitudinal direction, it can also excite the string in the transversal directions at the string termination point, provided that the termination is nonrigid in the transversal plane. Figure 2.4 clarifies this.

As can be seen from the illustration, the varying of the tension, called *tension modulation driving force* (TMDF) [60], excites the bridge in the vertical direction. If the bridge has a nonzero mechanical admittance in the vertical direction, TMDF will cause the bridge to move up and down.

Let us now consider a case where a vibrating string carries two transversal modes, n and m. As stated above, the TM caused by mode n has a sinusoidal form with a frequency corresponding to mode $2n$. When this vibration is coupled through the nonrigid bridge with the transversal vibration of mode m, the resulting vibration is the mode $2n$ amplitude-modulated with mode m. Thus, this vibration can excite mode p, if $p = |2n \pm m|$.

Now it is clear that even if a string is plucked at $x = p/L$, the pth mode, although supposed to be missing, will receive energy from other modes because of this nonlinear coupling at the

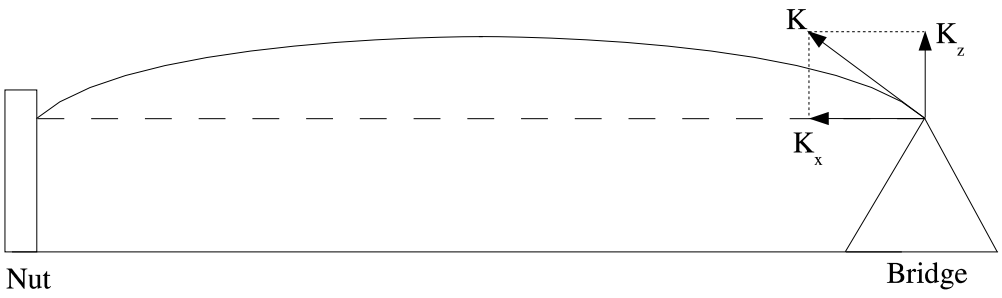

Figure 2.4: Tension modulation exerts a transversal force K_z on the bridge. If the bridge is able to move in the z-direction, this vibration will be coupled with the string. With a rigid bridge, no coupling will take place, and the missing harmonics will not be generated (after [39]).

bridge. It is important to note, however, that this phenomenon cannot excite all the modes, e.g. if the string is plucked near its center so that no even modes will be present, they will not be generated by this mechanism either.

Since this energy transfer from other modes is rather gradual than instantaneous, the missing modes excited by the TMDF will experience a gradual onset, and behave like other modes after reaching their peak values. The rising rate of these missing harmonics can be shown to be proportional to the cube of the pluck amplitude [39].

In real musical instruments, also another mechanism is responsible for the generation of missing harmonics. The string is often terminated behind the bridge, and it undergoes a change in direction at the bridge location. Figure 2.5 illustrates this fact. Now the TM can be directly coupled with the vertical polarization, due to angle α at the bridge. This means that the TMDF due to a transversal mode n will have a frequency corresponding to mode $2n$, so this mechanism can excite only even modes, thus rising the even harmonics also in a middle-plucked string [39].

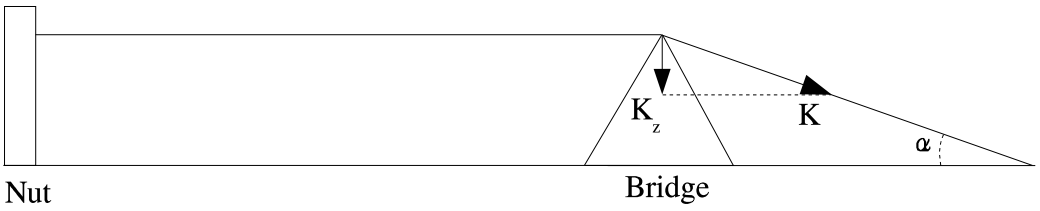

Figure 2.5: A more realistic bridge model. The angle α causes the tension to have a vertical component, K_z. This results in a TMDF with frequency twice as high as the transversal frequency in the string (after [39]).

2.2.4 Other nonlinearities

Although this thesis is mainly concerned in the nonlinearity due to tension modulation, there exists also other important sources of nonlinearity in the realm of string instruments. Some of these are discussed briefly in the following.

The vibrational amplitude of a plucked string is sometimes limited by the instrument body itself. For example in guitars, the fretboard, which gives the performer the freedom of playing different notes, also acts as a limiter for the string displacement. This limiting phenomenon can also be used as a special effect, as is often done in playing the electric bass guitar. In this playing technique, commonly known as the *slap bass*, the player hits the string usually with the side of her thumb, allowing the string to "slap" on the fretboard. The string displacement in a case like this is said to be *hard-limited*. We will discuss the modeling of the hard-limiting nonlinearity in the following chapters.

A plucked string instrument, as its name implies, is excited using some kind of a tool to displace the string and then release it. Usually, the excitor used is a finger or a *plectrum*, a small somewhat elastic piece of plastic or other material. Since neither of these excitors are totally rigid, but also possess elasticity, the type of the string release is rather gradual than instantaneous. This phenomenon, commonly referred as the string-finger contact, introduces nonlinearity to the excitation, and thus also to the following string vibration.

Other types of interesting nonlinear string excitations include the hammer-string contact in piano-like instruments and the bow-string interaction in bowed string instruments. These phenomena fall out of the scope of this thesis. The hammer-string nonlinearity is mainly due to the compression of the hammer felt, and it is covered in many earlier studies [23], [24], [25], [11], [59], and [9], to name a few. The bow-string nonlinearity is mainly caused by the stick-slip contact between the string and the bow. Also this interaction is covered in many studies, [40], [69], [50], and [16] present some of them.

Chapter 3

String Model Using Digital Waveguides

3.1 Synthesis Through Waveguide Modeling

Digital waveguide modeling is a term often encountered when studying the synthesis of string instruments. It is based on the fact that when an exciting signal is inserted into a string, it is reflected at the boundaries, and returns to its initial position. At its simplest form, this can be implemented as a single delay-loop with two consecutive samples averaged, as is done in the classic *Karplus-Strong* (KS) algorithm [36]. The averaging corresponds to lowpass-filtering and thus simulates the frequency dependent losses in the string. A noise burst is usually used as the exciting signal in the most basic KS synthesizer. The basic Karplus-Strong algorithm is illustrated in Fig. 3.1. It can produce surprisingly high quality sounds despite its computational simplicity.

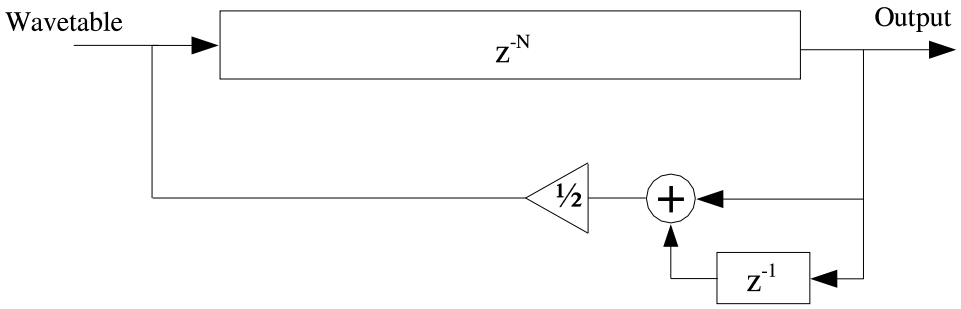

Figure 3.1: The classic Karplus-Strong algorithm. The averaging of two consecutive samples corresponds to the lowpass-effect of the frequency-dependent losses. The excitation signal for the string model is read from a wavetable (after [36]).

More sophisticated string instrument models can be obtained by extending the KS model for

example by changing the averaging filter into a higher degree one, or by filtering the output of the string model with a filter simulating the body response of the instrument. A tuning filter can also be added to enable correct tuning of the string models, a problem which will be discussed more throughly in Section 3.2.2. Also, a more realistic excitation signal can be obtained by inverse filtering a recorded string tone with a corresponding KS model.

The computational load of the model can be reduced further, if the body response effects are included in the excitation signal. This is generally known as *commuted waveguide synthesis* (see e.g. [34]), and it can be used to produce very realistic plucked-string instrument tones. A fine overview of extended KS models and DWG's can be found in [35].

The string models discussed so far in this chapter comprise from a single delay loop with predefined input and output locations. Since the various signal processing blocks inside these models are lumped rather than distributed, it is not possible to gather meaningful data inside these structures, but only from a certain predefined output location. In the following, models without this constrain are discussed.

3.2 DWG Model of an Ideal String

This section describes the basics of DWG modeling and presents a computational model of an ideal string. An excellent introduction to DWGs used in modeling musical instruments can be found in [56].

3.2.1 Traveling-wave solution in the digital form

The traveling-wave solution (Eq. 2.5) of the 1D wave equation (Eq. 2.1) can be converted into digital form by sampling the wave components at T intervals. Usually also the spatial sampling interval X is chosen so that the wave propagation velocity satisfies $c = X/T$. This causes the wave components to propagate one spatial sample in one time step. The sampling is done formally by changing the variables in Eq. 2.5 [56]

$$x \rightarrow x_m = mX$$
$$t \rightarrow t_n = nT.$$

If now the traveling waves are redefined as

$$y_l(n) = y_l(nT)$$
$$y_r(n) = y_r(nT)$$

the discrete traveling-wave solution can be obtained

$$y(t_n, x_m) = y_r(n - m) + y_l(n + m). \tag{3.1}$$

The term $y_r(n-m)$ can be thought as $y_r(n)$, delayed by m samples. Similarly, the term $y_l(n)$ can be thought as $y_l(n+m)$, delayed by m samples. It is important to note that Eq. 3.1 is *not* a mere approximation of Eq. 2.1 , but yields *exact* results at the sampling instants, within the limits of the numerical precision of the samples [56]. This kind of structure can be easily implemented with two delay lines containing unit delays, and $y(t_n, x_m)$ can be obtained by summing the delay line values at correct locations. The string state at the next time step can be updated by simply shifting the samples one step in the direction of the delay line. This procedure is shown in Figure 3.2.

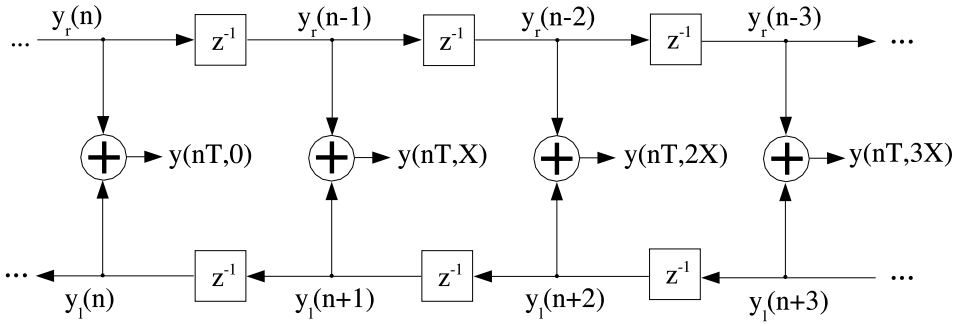

Figure 3.2: A dual-delay line model of an ideal string. The state of a string segment can be obtained at any spatial location by summing the values of the two delay lines at the same location. Note that both delay lines are one-directional (after [55]).

Clearly, this kind of structure allows more freedom in choosing the output location than the previously presented Karplus-Strong model. This model can be called *spatially distributed*, since we can gather meaningful data from various locations on this string model. We can also excite this model at various locations by means presented later in this chapter. This is an important feature, since especially in string instruments, the excitation location determines the resulting spectrum, as shown in Chapter 2. Also, the output location determines the resulting sound by some extent as can be seen in electric guitars, where similar pickups at different locations yield different tones.

In order to model the wave reflection at termination points of an ideal string, a sign change must be introduced at the ends of the delay lines. This simulates the phase-inverting reflection at the string end. Figure 3.3 shows a DWG model of an ideal string with proper terminations. In this figure, the delay lines are illustrated so that only one movable output location is shown, thus allowing the output point to be located anywhere on the string [1].

[1]The possible output locations have integer indices by nature. If a noninteger output location is desired, spatial interpolation or fractional delay elements must be used. The latter topic will be discussed in more detail in Section 3.5.1.

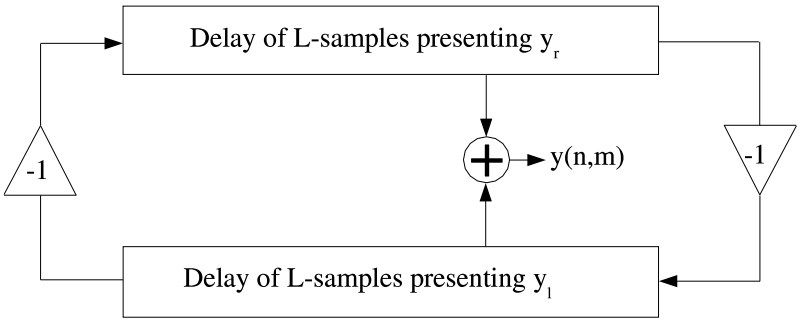

Figure 3.3: A DWG model of an ideal string. The unit delays in Fig. 3.2 are combined here for illustrational purposes to produce a bidirectional delay line. The phase-inverting reflection at the termination points is implemented with a sign change (after [55]).

3.2.2 Tuning filters

It is obvious that when a string is modeled as a delay line its fundamental frequency depends on the length of the delay line. More specifically, if the wave components propagate one spatial sample in one time step, the fundamental frequency of the string model can be presented as $f_1 = f_s/(2L)$, where f_s is the sampling rate and L is the delay line length. Now if the delay lines have integer lengths, the frequencies can have only certain values, provided that the sampling rate is constant. This means that the frequency scale is not stepless, and realistic tuning of the string models becomes impossible, especially at high frequencies. This is illustrated in Figure 3.4.

This problem can be solved by inserting a special digital filter into the delay-line loop discussed above [30]. Such a filter is able to produce also *fractional delay* (FD) values, and is therefore called a *fractional delay filter* [37]. We preserve a more thorough discussion of the FD filters for Section 3.5.1.

3.3 DWG Model of a Lossy String

The losses in a vibrating string, discussed in Section 2.1.4, are often simulated in computational models using *frequency-independent* and *frequency-dependent* losses. The former is equivalent with the first loss term of Eq. 2.7, namely the term containing the first temporal derivative. It can be checked that, when modeling the frequency-independent losses, d'Alembert's solution can be approximated as [56]

$$y(t,x) = e^{-(d_1/2\epsilon)x/c} y_r(t - x/c) + e^{(d_1/2\epsilon)x/c} y_l(t + x/c), \tag{3.2}$$

provided that the string displacement is small. If these exponentially decaying traveling waves are spatially and temporally sampled (using $X = cT$, as before), we end up with the discrete

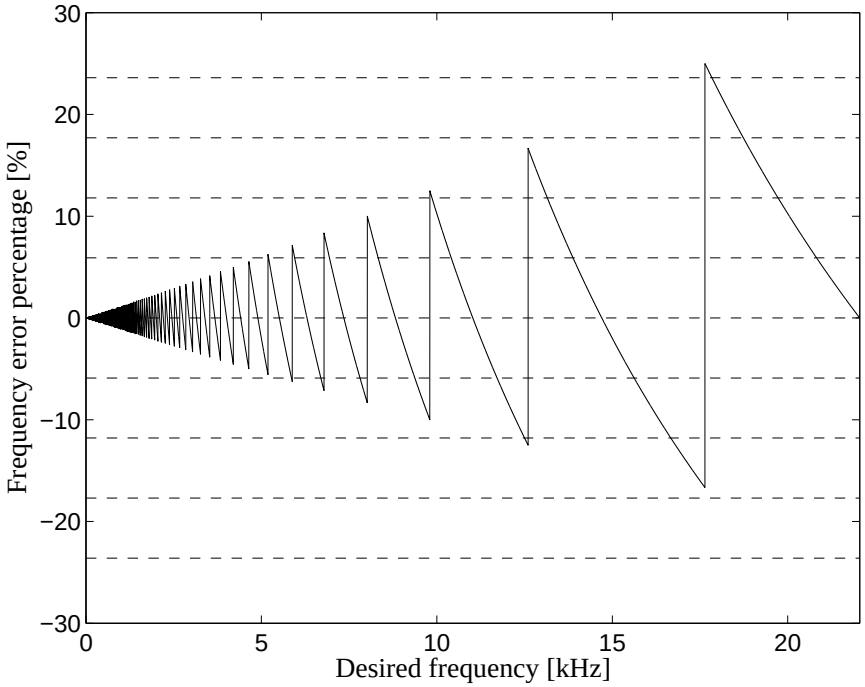

Figure 3.4: Tuning error as a function of the fundamental frequency for a DWG with integer delay values. The sampling frequency is 44.1 kHz. The tuning error is drawn with a solid line, while the dashed lines denote semitone steps of the musical scale. For reference, it has been shown [41] that humans are able to distinguish frequency errors even below 0.5 %.

traveling wave solution with frequency-independent losses [56]

$$y(t_n, x_m) = g^{-m} y_r(n-m) + g^m y_l(n+m), \tag{3.3}$$

where $g = e^{-d1T/2\epsilon}$.

The implementation of Eq. 3.3 is straightforward, since only a constant gain element must be added in the lossless DWG implementation. However, the precise modeling of frequency-dependent losses in DWG's is a far more demanding task. Instead of trying to formulate the remaining loss term in Eq. 2.7 as additional terms in Eq. 3.2, the frequency dependent losses are often modeled using a *loss filter* (a.k.a. *loop filter*), which is inserted in the delay-line loop. This element is usually a lowpass-filter, since high frequencies attenuate faster in the vibrating string. A lossy DWG string containing both frequency-dependent and frequency-independent losses is illustrated in Fig. 3.5.

This approximation of the frequency dependent losses without a rigorous derivation of terms from the lossy wave equation should not seem too harsh a maneuver due to the LTI assumption discussed in Chapter 2, and also since Eq. 2.7 is an approximation of the real-world case to begin with.

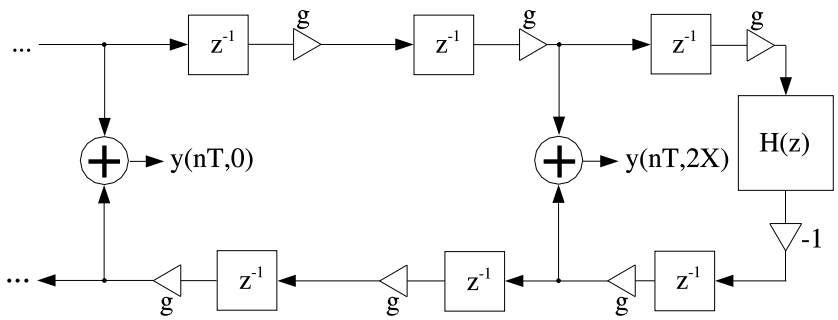

Figure 3.5: A lossy DWG string. The frequency-independent losses are implemented with a gain factor $0 \leqslant g \leqslant 1$, and frequency-dependent losses with a loss filter $H(z)$ (after [55]).

3.3.1 Effects of stiffness in DWG's

The dispersive behavior of stiff strings can be simulated in DWG models by inserting suitable allpass filters between the unit delays. The allpass filters do not change the amplitudes of the wave components but introduce different phase delays for different frequencies, thus simulating dispersion. Often in practical implementations the allpass filters are lumped as one single filter at the termination (since allpass filters are generally LTI), in a similar way that was done with the frequency dependent losses, in order to reduce the computational load. The usage of allpass filters in simulating the string stiffness is discussed for example in [56].

3.4 Implementing Nonlinearities in DWG Models

When the hammer-string nonlinearity, briefly mentioned in Chapter 2, is implemented in a DWG structure, problems arise due to the feedback loop between the string and the hammer and spatial discontinities caused by discrete sampling. The contact between the hammer and a DWG string has been studied for example in [3] and [6]. Another interesting nonlinear type of string excitation is the bow-string contact. It has been studied in the DWG context previously in [56], [57], and [28], for example.

The hard-limiting nonlinearity, also discussed in Chapter 2, can easily be modeled in DWG systems. The key idea here is [32] that if the string displacement at a certain point exceeds the constrained value, a negative impulse excitation is inserted to the same location. The amplitude of the impulse should be chosen so that the displacement will have the maximum allowed value. Modeling of the tension modulation in DWGs is discussed more thoroughly in the following.

3.5 Tension Modulation in DWG's

When tension modulation is to be implemented in a string model, it clearly means that the fundamental frequency of the string must be modulated also (see Eq. 2.2). In a DWG string this corresponds to varying either the length of the delay lines or the temporal sampling instant. This section discusses implementing the TM by varying the length of the delay lines, and is presented earlier in recent publications by the author [42] and [43]. Since the delay lines can generally have integer-valued delays only, directly altering the delay lines would lead to having the tension change in a stepwise manner. Obviously, this behavior is not desired, thus we need a tool for implementing delays with arbitrary lengths. This kind of tool is called a fractional delay filter, and it is shortly discussed in the following. For an excellent tutorial about FD filters, see [37].

3.5.1 Fractional delay filters

Basically, a fractional delay filter is a device for bandlimited interpolation between samples. The key idea is to design a filter which will apply a desired amount of delay to the signal without altering it otherwise. This means that the filter should have flat phase delay and magnitude responses, and it should be able to have a phase delay of any value. Unfortunately, this ideal solution is impossible to implement in reality, and all FD filters are more or less approximations of the ideal case. The nonideality will manifest itself especially at high frequencies, as we are about see.

In practice, the FD elements can be implemented with FIR or IIR filters. The FIR filters, although generally easier to design, will always have a nonflat magnitude response at high frequencies. This is not the case with IIR filters, since they can be implemented with an allpass structure. Both filter types will, however, have nonideal phase responses, where the "phase error" tends to increase towards the high frequencies. A first-order allpass filter was chosen for the FD element of our string model. Its transfer function is of the form

$$A(z) = \frac{-a + z^{-1}}{1 - az^{-1}} \tag{3.4}$$

where a is the filter coefficient which defines the length of the delay. Figure 3.6 illustrates the phase delay responses of the first-order allpass filter. Notice that when $a = 0$, the filter acts as a unit delay.

The decision for using a first-order allpass filter within the string model was made partly because it is the simplest way to design an allpass filter approximating a given fractional delay [37], and partly because the first-order allpass filter is the best choice for the fractional delay element when delay values around unity are to be obtained [65]. The phase response error caused by the allpass filter is not considered to pose a problem since its effect is negligible in the audio frequency range, assuming the sampling frequency to be reasonably high [67].

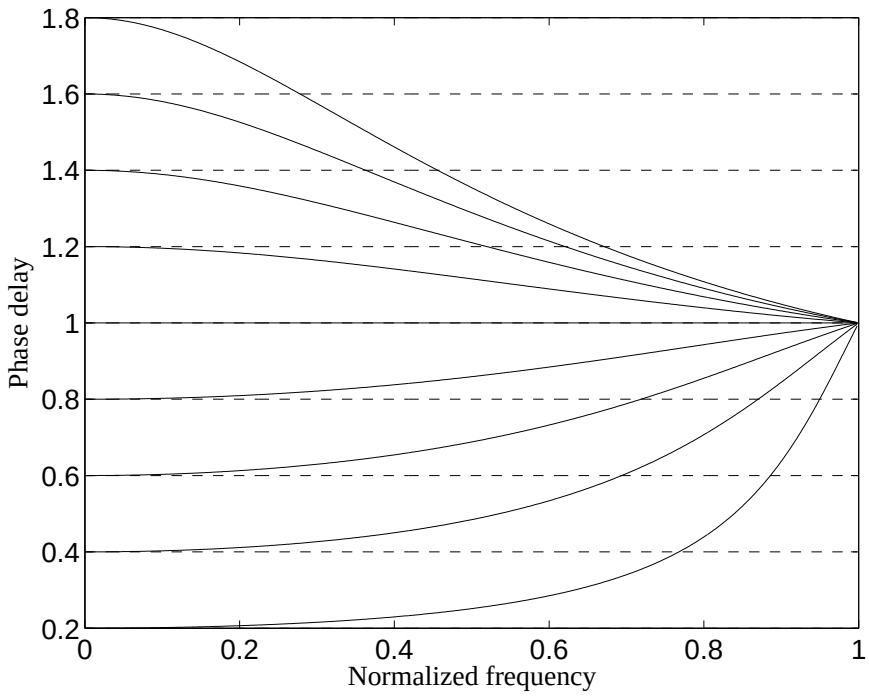

Figure 3.6: Phase delay response of a first-order allpass filter. The solid curves represent the phase delay responses at different delay values, starting from $d = 0.2$ up to $d = 1.8$. As can be seen in the figure, the obtained delay values (solid lines) coincide well with the desired delay values (dashed lines) only at low frequencies. Clearly the filter is ideal only when $d = 1$, i.e. when it produces a unit delay.

3.5.2 Distributed nonlinear DWG string

Previous works [66], [61], [19] use a single fractional delay element in a single-polarization string model, or a DWG string terminated with a nonlinear double-spring [48] to simulate the nonlinear string. This is done in order to reduce the computational complexity of the model, but it has also some shortcomings. Since the system does not meet the LTI requirement[2] , the FD elements cannot be lumped into one single element without giving up the idea of viewing the system as a distributed model. In other words, the whole string becomes a lumped model, and the termination point "behind" the FD element becomes the only location for gathering a meaningful output signal from the string. Physically this would correspond to a single elastic element at the termination point of an otherwise rigid string. A more realistic solution can be obtained if the elongation process is distributed along the delay line, in a way similar to a real physical string, where the elasticity is distributed along the string, rather than lumped.

[2]More specifically, varying the allpass filter coefficients makes the system time-variant, and the fact that the system state (string elongation) defines the allpass filter values makes the system nonlinear. Clearly, the actual physical phenomenon is both nonlinear and time-invariant.

The distributed nonlinearity can be implemented by exchanging every other unit delay element in the DWG model (Figures 3.2 and 3.5) with a first-order allpass filter. Then the effective length of the delay lines can be changed by varying the filter coefficients. We will now introduce a delay block which contains a unit delay, a first-order allpass filter, and two scaling coefficients for modeling the frequency-independent losses. This block is called a *basic element*, and it is illustrated in Figure 3.7(a). Figure 3.7(b) shows how to obtain output data from a junction between two basic elements. The nonlinear DWG string can now be constructed from the lossy DWG structure in Fig. 3.5 by replacing each unit delay pair with a basic element. The resulting structure is illustrated in Figure 3.8.

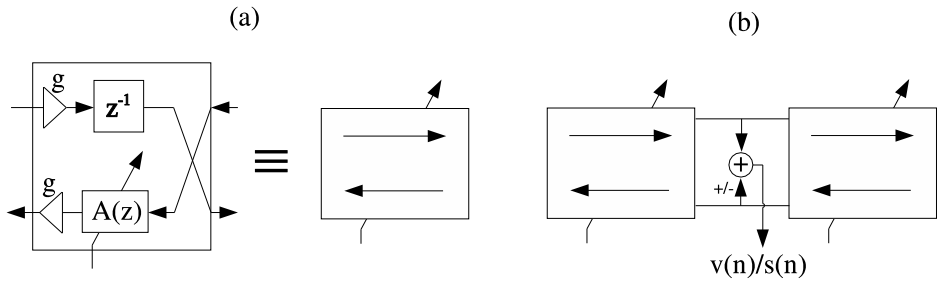

Figure 3.7: Illustration of (a) a basic element, and (b) how to obtain output data from a string consisting of these elements. The directions of the wave components in (a) are opposite for adjacent elements. In (b), either the velocity or the slope of the string segment can be obtained, if velocity is used as the wave variable.

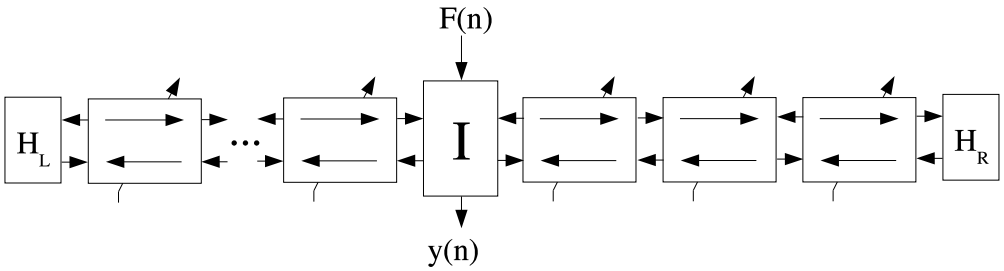

Figure 3.8: One-polarizational nonlinear DWG string. The string consists of the basic elements illustrated in Fig. 3.7(a). H_L and H_R denote the loop filters simulating the frequency-dependent losses. The excitation to the string can be inserted as a force signal using an *interaction element*, denoted by I. The construction of the interaction element is illustrated in Figure 3.9.

From the discussion in Section 2.2.1, we can conclude that the suitable control signal for the FD elements can be derived from the elongation of the string. In the following, since $c_t \ll c_l$, we assume that the longitudinal waves will propagate instantaneously through the string, and

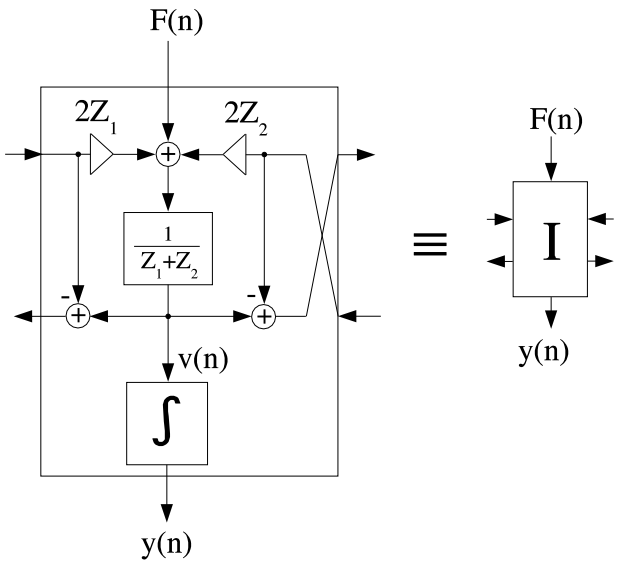

Figure 3.9: The interaction element allows excitation signals to be inserted to the string during run-time. The input signal $F(n)$ can be seen as a force signal, and the output signal $y(n)$ as a displacement signal. The coefficients Z_1 and Z_2 represent the mechanical impedances of the two string branches. Implementation of the integrator block is depicted in Figure 3.10.

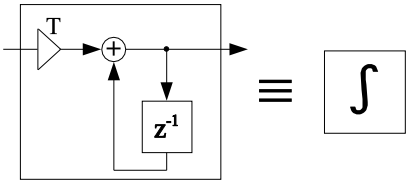

Figure 3.10: The integrator block is implemented by summing up consecutive samples.

the elongation calculation and the FD parameter tuning can be done for the whole string in one piece. Strictly speaking, of course, this is not the case with real strings, but carrying out the FD parameter evaluations for multiple string segments would add a significant computational load, likely without any audible advances. The elongation of the string can be expressed as [39]

$$l_{dev}(t) = \int_0^{l_{nom}} \sqrt{1 + (y_x(t,x))^2}dx - l_{nom} \tag{3.5}$$

where l_{nom} is the nominal string length, x is the spatial coordinate along the string, and y is the displacement of the string. The first spatial derivative in equation (3.5) suggests the use of slope waves in the elongation calculation, and thus equation (3.5) can be approximated for the digital waveguide as [61]

$$L_{dev}(n) = \sum_{m=0}^{\hat{L}_{nom}} \sqrt{1 + [s_r(n,m) + s_l(n,m)]^2} - \hat{L}_{nom} \tag{3.6}$$

where $s_r(n,m)$ and $s_l(n,m)$ are the slope waves at time instant n and position m, propagating to the right and to the left, respectively. $\hat{L}_{nom}$ is the rounded nominal string length. To reduce the computational complexity, equation (3.6) can be further simplified using a truncated Taylor series expansion to [61]

$$L_{dev}(n) \approx \frac{1}{2} \sum_{m=0}^{\hat{L}_{nom}} [s_r(n,m) + s_l(n,m)]^2 \tag{3.7}$$

while still maintaining a sufficient accuracy. The approximated delay variation of the total DWG can be obtained from equation (3.7) as [61]

$$D_{tot}(n) \approx -\frac{1}{2} \sum_{l=n-1-\hat{L}_{nom}}^{n-1} (1 + \frac{EA}{K_0}) \frac{L_{dev}(l)}{L_{nom}} \tag{3.8}$$

where E is Young's modulus, A is the cross-sectional area of the string, and K_0 is the nominal tension corresponding to the string at rest.

Since the system under consideration uses a distributed set of delay elements, the total delay must be divided by the number of delay elements in the DWG to obtain a delay value for each individual element. The coefficient a in Eq. 3.4 can now be expressed as [67]

$$a = \frac{1 - d_{partial}}{1 + d_{partial}} \tag{3.9}$$

where $d_{partial}$ is the delay intended for a single allpass filter. Note that previous studies have used a different sign for a in Eqs. 3.4 and 3.9, although the operation of the allpass filter remains the same.

3.5.3 Stability of the nonlinear digital waveguide algorithm

Although the stability of the nonlinear DWG algorithm can not be guaranteed in theory, we found it experimentally to remain stable for nearly all parameter and excitation values. Only highly exaggerated nonlinearities together with high excitation impulses resulted in stability problems. We thus conclude that the nonlinear DWG waveguide has no real stability problems when synthesis of natural plucked-instrument sounds are desired.

3.6 DWG Model of a Kantele

In this section, we demonstrate the nonlinear DWG formulation by constructing a two-polarizational synthesis model of a *kantele*, a Finnish folk music instrument.

3.6.1 Acoustical analysis of the kantele

The kantele is a bridgeless plucked string instrument, with usually five metal strings in its basic form (see Fig. 3.11). The strings are terminated at one end by metal tuning pins, which are screwed directly into the soundboard. At the opposite end all strings are wound once around a horizontal metal bar called the varras and then knotted. Because of the nonzero distance between the center of the varras and the knot, the vibrations in two polarization planes have different effective lengths; the varras is the termination point for horizontal vibration, while the knot acts as the termination for the vertical vibration, as illustrated in Fig. 3.11. This phenomenon causes the total vibration of the string to have two fundamental components with slightly different frequencies, producing beating [19]. A more detailed structure and acoustical

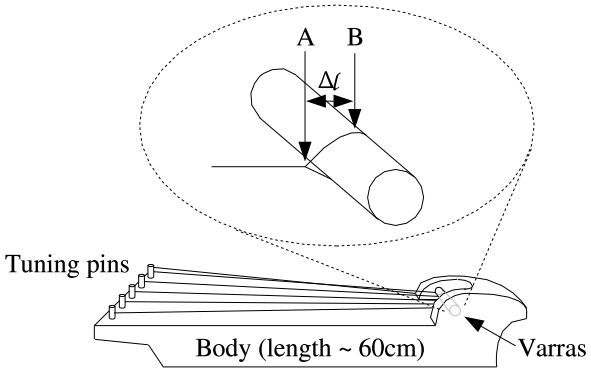

Figure 3.11: Illustration of the kantele. The string termination at varras is magnified for clarity. A denotes the termination point for vertical vibration of the string, while B denotes the termination point for horizontal vibration. $\triangle l$ stands for the distance between A and B.

analysis of the kantele can be found in [19]. A study of the history of kantele and an acoustically improved new design are presented in [52].

3.6.2 A novel kantele string model

The novel synthesis model of a single kantele string is constructed using two single-polarization nonlinear DWG models, illustrated in Fig. 3.8, and connecting them together via a scaling coefficient for modeling the coupling between the two polarizations. We restrict the coupling to being one-directional in order to avoid stability problems which would otherwise rise due to the feedback loop formed by the interconnected strings, as suggested by Karjalainen et. al. [35]. The elongation approximations of the strings and the resulting allpass filter coefficient values are evaluated separately for the two DWG models using the arithmetics described in Sec. 3.5.2. The structure of the novel kantele string model is illustrated in Figure 3.12. In this model, $f_{in}(n)$ denotes the plucking force exciting the string and $v_z(n)$ and $v_y(n)$ represent the velocity signals coming from the strings vibrating vertically and horizontally, respectively.

It is important to note that while $v_z(n)$ and $v_y(n)$ can be obtained anywhere along the string, in this case they are evaluated at the termination points, so that terminal impedances can be used. $Z_{zbridge}$ and $Z_{ybridge}$ represent the vertical and horizontal terminal impedances, respectively. Z_c stands for the coupling impedance from vertical to horizontal string vibration polarization, and $f_{zy}(n)$ represents the corresponding driving force. The forces to the termination caused by the two one-polarizational vibrations are denoted by $f_z(n)$ and $f_y(n)$. The connection from the elongation approximation block to the output simulates the direct coupling of the TM to the instrument body [60]. The force signal $f_x(n)$ represents the TM modulation signal, and a scaling coefficient, denoted by C, is used to control the amount of this coupling. The output of the whole two-polarization string model is finally presented as a force signal $f_{out}(n)$ excerted to the string termination point. It must also be noted that this model simulates only a single kantele string, and a model of the instrument body must also be added, if realistic sound synthesis is desired.

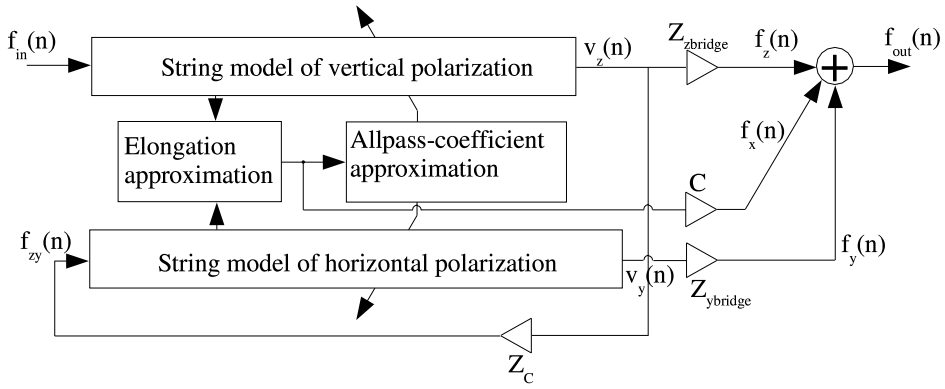

Figure 3.12: A kantele string model. The one-polarization string model blocks are identical to what is illustrated in Figure 3.8, but have different string lengths. It is important to note that the coupling between the vibrational polarizations in a real physical system is more complicated but a simplified one-way coupling is used here for ease of simulation.

3.6.3 Simulation results and comparison to measured data

The synthesis results reveal that the behavior of the nonlinear string model is consistent with the theory of elastic strings, discussed in Section 2.2. The fundamental frequency behaviors of a single-polarization string model and a real kantele tone are compared in Figure 3.13. The real kantele sound was recorded in an anechoic chamber using a moderately-plucked five-stringed kantele. In this figure, the horizontal dotted line approximates the mean value of perceptual detection threshold of an initial pitch glide. The psychoacoustic detection threshold in the frequency region of these tones is about 5.4 Hz [31]. This shows that the fundamental

frequency glide is an audible phenomenon in the kantele even at modest plucking amplitudes, and thus it must be modeled in the synthesis phase if high quality tones are desired. The string model used in this figure has a total delay line length of 55.125 and the allpass coefficient a is scaled using a constant value of 0.9 in order to simulate the behavior of the recorded sample. The sampling frequency is 44.1 kHz.

The modeling of the generation of missing harmonics can be implemented similarly in the distributed nonlinear DWG model as was suggested in [61]. If the boxcar integration of Eq. 3.8 is replaced with a leaky integrator having the transfer function

$$I(z) = g_p \frac{1 + a_p}{1 + a_p z^{-1}}, \tag{3.10}$$

the generation of the missing harmonics can be controlled via the integration parameter a_{p}. The variable g_{p} defines the gain of the integration.

Figure 3.14 shows the amplitude envelopes of the first three harmonics of a synthesized tone with two different a_p parameter values. The string is plucked in both cases using the same excitation signal applied at the same location, approximately $1/3$rd of the string's length, and as can be seen in the figure, the missing harmonic in (a) has a gradual increase after the beginning transient, after which it experiences an exponential decay like all other modes. The general form of the third harmonic in (b) is the same as in (a), but it is heavily attenuated. The string in both cases consists of 60 basic elements, and it is plucked between the 20th and 21st elements. We use two identical third-order lowpass FIR filters with coefficients $[-0.0732,\ 0.5668,\ 0.5668,\ -0.0732]$ at both ends of the string for modeling the frequency-dependent losses. The beating discussed in Section 3.6.1 is naturally absent in Figures 3.13 and 3.14, since only a single-polarization string was studied.

It is worthwhile to note that the generation of missing harmonics in the nonlinear DWG model results from the properties of the integration of the elongation approximation, and is therefore not a physically justified process. Also, unlike the real physical phenomenon, the generation of missing harmonics in the nonlinear DWG case does not depend on the rigidity of the terminations. Nevertheless, as can be seen, this feature can be exploited in emulating the real string behavior, when the integration parameters are properly adjusted. Details on tuning the leaky integrator parameters can be found in [61].

A real-time sound synthesis model of a kantele is constructed using a block-based, efficient audio DSP tool, the BlockCompiler. The algorithm used is efficient enough to process a five-string kantele model on an ordinary laptop computer at a 44.1 kHz sampling rate. A detailed description of the BlockCompiler is presented by Karjalainen [33].

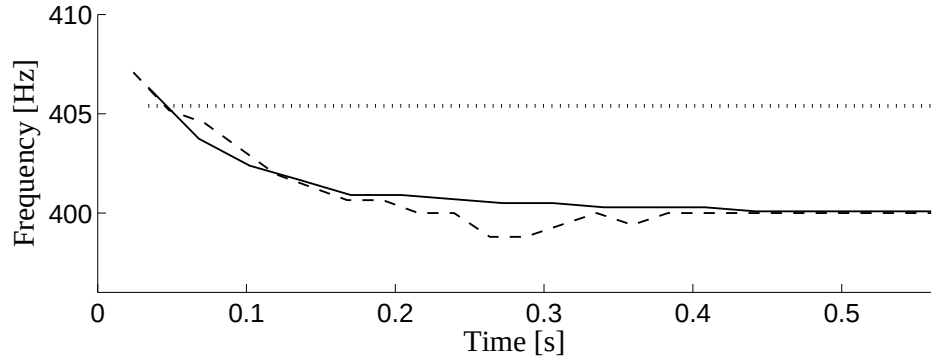

Figure 3.13: Illustration of the fundamental frequency glide in the synthesized tone (solid line) and in a real kantele tone, obtained via measurements (dashed line). The approximated detection threshold of a pitch drift, illustrated as a dotted line, suggests that the fundamental frequency drifts in both cases are audible.

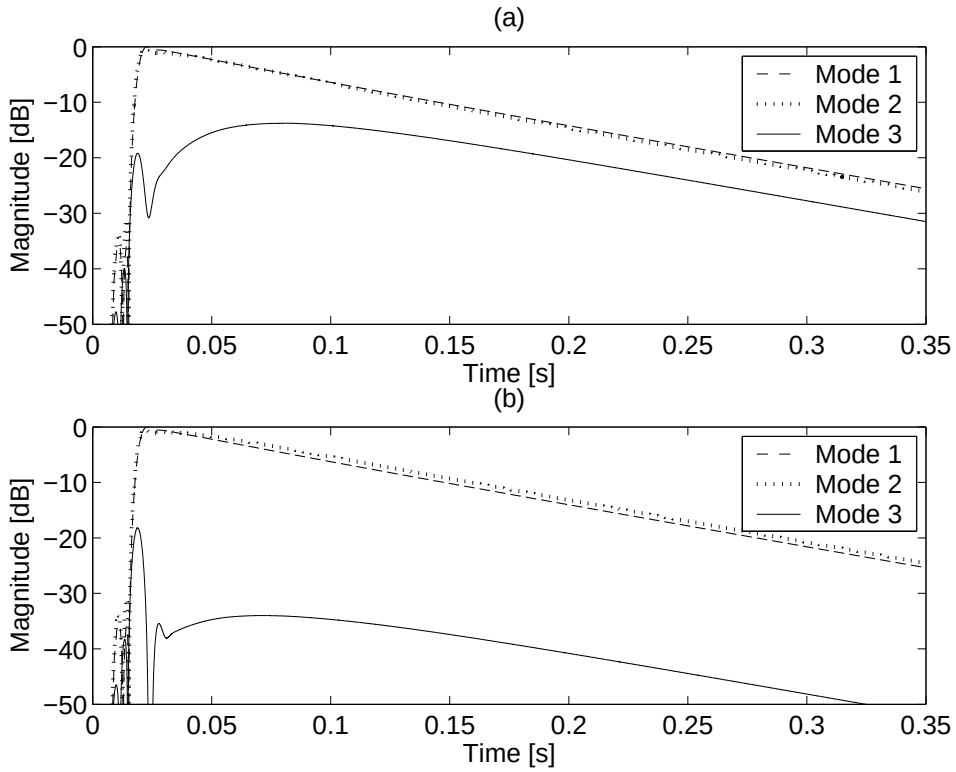

Figure 3.14: Generation of the missing harmonics in the nonlinear DWG model can be controlled via the leaky integrator parameters. Here, the string was plucked approximately at $1/3$rd of its length, so every 3rd harmonic should be missing from the resulting spectrum. In a), $a_p = -0.2902$ and the third harmonic clearly rises after the initial transient. In b), $a_p = -0.9672$ and the third harmonic is more attenuated.

Chapter 4

Nonlinear String Model Using Finite Differences

4.1 Synthesis Through Finite Differences

In the previous chapter we discussed string synthesis via discretizing d'Alembert's solution to the 1D wave equation. Another approach is to discretize the wave equation itself, for example by substituting finite difference terms for the derivatives in the wave equation (Eq. 2.1 in the case of an ideal string). This mode of operation is commonly known as the finite difference method, and it was first used for sound synthesis purposes by Hiller and Ruiz in the early seventies [26] and [27]. Finite differences had already earlier been used in mathematics for numerical solving of partial differential equations. A fine introduction to FDM in the synthesis of plucked stringed instruments can be found in [10].

In this chapter we will first review the ideal FDM string model and discuss its tuning and numerical dispersion. Next, we will have a look at the boundary conditions, review possible excitation schemes, and discuss the lossy FDM string. Finally, we will turn our interest to nonlinear FDM string models and present a novel algorithm for modeling nonlinear strings in Section 4.4.1.

4.2 Finite Difference Model of an Ideal String

The partial derivatives in the 1D wave equation (Eq. 2.1) can be replaced by finite differences:

$$y_t(t,x) \approx \frac{y(t,x) - y(t-T,x)}{T} \tag{4.1}$$

and

$$y_x(t,x) \approx \frac{y(t,x) - y(t,x-X)}{X}. \tag{4.2}$$

The second order temporal differentiation now becomes

$$y_{tt}(t,x) \approx \frac{y(t,x) - 2y(t-T,x) + y(t-2T,x)}{T^2}. \tag{4.3}$$

Next, a one-sample time shift is applied in order to avoid delay errors in the result, leading to

$$y_{tt}(t,x) \approx \frac{y(t+T,x) - 2y(t,x) + y(t-T,x)}{T^2}. \tag{4.4}$$

Likewise, for the spatial differentiation

$$y_{xx}(t,x) \approx \frac{y(t,x+X) - 2y(t,x) + y(t,x-X)}{X^2}. \tag{4.5}$$

After substituting Eqs. 4.4 and 4.5 in the 1D wave equation (Eq. 2.1), the 1D finite difference wave equation can now be written as

$$K\frac{y(t,x+X) - 2y(t,x) + y(t,x-X)}{X^2} = \epsilon\frac{y(t+T,x) - 2y(t,x) + y(t-T,x)}{T^2}. \tag{4.6}$$

Solving 4.6, we get

$$y(t+T,x) = \frac{KT^2}{\epsilon X^2}[y(t,x+X) - 2y(t,x) + y(t,x-X)] + 2y(t,x) - y(t-T,x). \tag{4.7}$$

Next, we define the relationship between the spatial and temporal sampling steps with [10]

$$r = c\frac{T}{X} \leqslant 1, \tag{4.8}$$

where the "less than unity" restriction is called the Von Neumann stability condition. We will call r a *tuning coefficient*[1] and justify the naming later. Using this together with the definition of transversal wave velocity (Eq. 2.2), Equation 4.7 can be written as

$$y(t+T,x) = r^2[y(t,x+X) + y(t,x-X)] + 2(1-r^2)y(t,x) - y(t-T,x). \tag{4.9}$$

If we now do the discretization by denoting $t \to t_n = nT$ and $x \to x_m = mX$, as we did in Section 3.2.1, we end up with the finite difference approximation [10]

$$y(n+1,m) \approx r^2[y(n,m+1) + y(n,m-1)] + 2(1-r^2)y(n,m) - y(n-1,m). \tag{4.10}$$

If $r = 1$, the finite difference approximations of Eqs. 4.1 and 4.2 become exact and the recurrence equation becomes

$$y(n+1,m) = y(n,m+1) + y(n,m-1) - y(n-1,m), \tag{4.11}$$

which is the finite difference equation of an ideal string. Equation 4.11 can conveniently be illustrated using a *spatio-temporal grid*, depicted in Figure 4.1. Here, the horizontal axis represents the spatial coordinate (i.e. position on the string) and the vertical axis represents the temporal coordinate (i.e. the time instant). We will find the spatio-temporal grid especially useful when illustrating the more complex nonlinear finite difference algorithm later.

[1] r is also known as the *Courant number*.

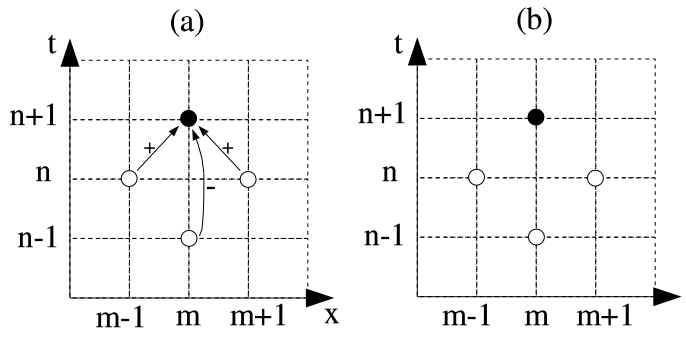

Figure 4.1: (a) Illustration of the 1D finite difference equation of an ideal string on a spatio-temporal grid. The vertical axis denotes the time, while the horizontal axis denotes the spatial location on the string. The left-hand side of Eq. 4.11 is presented with a black dot, while the values used in the right-hand side of Eq. 4.11 are presented as blank dots. The arrows in (a) denote the arithmetic operators of the equation and are usually left out, thus leading to a simplified form, illustrated in (b).

4.2.1 Numerical dispersion and tuning

The Von Neumann stability condition can also be expressed in the frequency domain by rewriting Eq. 4.8 in terms of sampling frequency $f_s = 1/X$ and fundamental frequency of the string $f_1 = c/2L_{nom}$ leading to [10]

$$N f_1 \leqslant f_s/2, \tag{4.12}$$

where N is the maximum amount of partials of the string tone, as stated by the Nyquist theorem. N also denotes the number of spatial points to be used in the simulation with sampling frequency f_s, if a string with fundamental frequency f_1 is desired. The fact that N is an integer now raises problems in the tuning of the string, since only discrete fundamental frequency values can be obtained. The solution for the tuning problem is to use $r < 1$. Since the variation domain of the fundamental frequency of the string f_1 is [10]

$$\triangle f_1 = \frac{f_s}{2L_{nom}} - \frac{f_s}{2(L_{nom}+1)} = \frac{f_s}{2L_{nom}(L_{nom}+1)}, \tag{4.13}$$

and since r depends linearly on f_1, the correct value of r for a given fundamental frequency f_1 can be derived. This is illustrated in Figure 4.2.

Choosing $r < 1$ also gives raise to an unwanted numerical dispersion phenomenon called *grid dispersion* [11], where the wave velocity in the numerical implementation will be less than the ideal physical wave velocity. This artificial dispersion affects primarily the upper harmonics, where the frequencies will be underestimated. If a typical error of 1% in the generated frequencies is allowed, the difference between the tuning coefficient r and unity should not be greater than 10^{-4} [10]. If the constraints between the correct tuning and grid dispersion do not

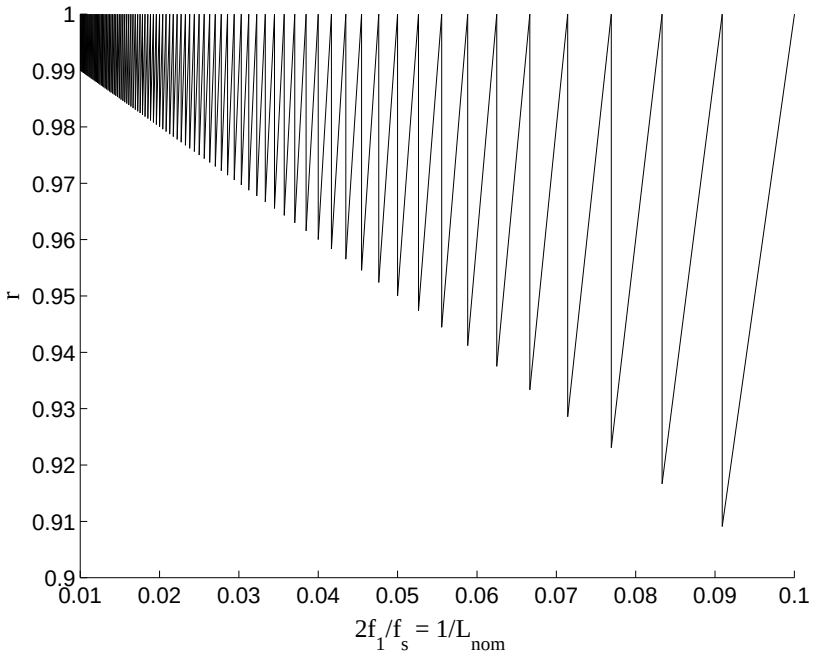

Figure 4.2: Tuning coefficient r as a function of normalized fundamental frequency. The string length L_{nom} lies within the range [10, 100] (after [10]).

yield satisfactory results, the spatial density of the grid should be increased. This is known as *spatial oversampling*.

Another clue that would suggest to use spatial oversampling is the fact that artificial damping is introduced in the high-frequency range by the discretization process of Eqs. 4.1 and 4.2. This happens because substituting the finite differences for the derivatives in the wave equation corresponds in frequency domain to mapping the $j\omega$ axis in the S plane (continuous) to the circle of radius $1/2$ centered at $z = 1/2$ in the z plane (discrete) [56]. The damping is negligible in the low frequencies, but as the frequency increases, the image of the $j\omega$ axis diverges away from the unit circle.

It must be noted that despite the artificial damping, Equation 4.11 yields still exact results at the sampling instants in the case of an ideal string. This can be proven [56] by substituting the discrete traveling-wave solution (Eq. 3.1) into the right-hand side of Eq. 4.11, which shows that Eqs. 3.1 and 4.11 are actually *equivalent*. From the discussion in Section 3.2.1, we conclude that Eq. 4.11 must then also yield exact results at the sampling instants

4.2.2 Boundary conditions and string excitation

Since the spatial coordinate m of the string must lie between 0 and L_{nom}, problems arise near the ends of the string when evaluating Eq. 4.11 because spatial points outside the string are needed. The problem can be solved by introducing boundary conditions that define how to

evaluate the string movement when $m = 0$ or $m = L_{nom}$. The simplest approach, introduced already in [26], would be to assume that the string terminations be rigid, so that $y(n, 0) = y(n, L_{nom}) = 0$. This results in a phase-inverting termination, which suits perfectly the case of an ideal string. A noninverting termination of a finite difference system is discussed in [32].

For a more realistic string termination model, several solutions have been introduced. In [26], Hiller and Ruiz formulate boundary conditions for a lossy termination including reflection coefficients r_l and r_r for the left and right termination, respectively

$$y(n+1, 0) = (1 - r_l)y(n, 1) + r_l y(n-1, 0) \tag{4.14}$$

and

$$y(n+1, L_{nom}) = (1 - r_r)y(n, L_{nom} - 1) + r_r y(n-1, L_{nom}). \tag{4.15}$$

If the supports are lossless, i.e. if there is a perfect reflection at the boundaries, r_l and r_r will equal unity; in general, however, they will be numbers between 0 and 1. For a guitar-like string termination, where the string is clamped just behind the bridge, it is assumed [10] that the string termination point and the bridge share the same displacement, leading to

$$y(n, L_{nom} - 1) = y(n, L_{nom}). \tag{4.16}$$

The excitation of an FDM string can also be implemented in several ways. Since our primary interest is focused on plucked-string instruments, we will not discuss violin- or piano-like excitations here. Probably the simplest way to excite an FDM string is to set its initial shape to match the displaced string prior to the release and then carry on with the simulation. If the initial shape $y(0, m)$ of the FDM string is set to resemble e.g. the triangular form of a plucked string, and if we assume that the string has no initial velocity, the string displacement at time instant $n = 1$ must be set to [26]

$$y(1, m) = \begin{cases} y(0, m), & m = 0, 1, ..., L_{nom}, \quad m \neq k, k+1 \\ \frac{1}{2}[y(0, m-1) + y(0, m+1)], & m = k, k+1, \end{cases} \tag{4.17}$$

where k and $k+1$ denote the coordinates of the slope discontinuity (i.e. the peak of the triangle). In a way, this points out the "correct direction" for the string vibration after the initialization. It must be noted, however, that if the spatial resolution of the string is high, the vertex of the triangle must be smoothed to avoid slope discontinuities that might cause problems with high frequency partials and grid dispersion, discussed in Section 4.2.1.

The plucking of a real string is, however, a more complex scheme than simply releasing the string from its initial shape. The properties of the finger or a plectrum alter the behavior of the string, and measurements reveal that the release is rather gradual than instantaneous [45], [46]. More advanced excitation models for FDM strings have been suggested e.g. in [10] and [18]. A simple but useful FDM string excitation method, which allows for interaction with the string

during run-time has been proposed in [32]. There,

$$y(n,m) \leftarrow y(n,m) + \frac{1}{2}u(n) \tag{4.18}$$

and

$$y(n,m+1) \leftarrow y(n,m+1) + \frac{1}{2}u(n) \tag{4.19}$$

are inserted into the string, which causes a "boxcar" block function to spread in both directions from the excitation point pair. Here, the impulsive wave component $u(n)$ can be used as the excitation signal in a similar way as the exciting force signal $F(n)$ in Section 3.5.2.

4.3 Finite Difference Approximation of a Lossy String

Frequency-independent losses can be modeled in an FDM string by discretizing the velocity-dependent damping term in the lossy 1D wave equation (Eq. 2.7). The first-order temporal derivative can be approximated using the *central difference* scheme, which gives [10] [18]

$$y_t(t,x) = \frac{y(n+1,m) - y(n-1,m)}{2T} + O(T^4), \tag{4.20}$$

where $O(\cdot)$ denotes the order of the approximation error and T is, as before, the temporal sampling interval. The reason for using the central difference scheme is its smaller approximation error compared to other schemes [58]. Inserting the central differences to the frequency-independent lossy 1D wave equation (Eq. 2.7 with $d_2 = 0$) yields [10]

$$y(n+1,m) = g_m(y(n,m+1) + y(n,m-1)) - a_m y(n-1,m), \tag{4.21}$$

where

$$g_m = \frac{1}{1 + d_1 T/2} \tag{4.22}$$

and

$$a_m = \frac{1 - d_1 T/2}{1 + d_1 T/2}. \tag{4.23}$$

The string decay time can now be directly controlled via the damping coefficient d_1, since $d_1 = 1/(2\tau)$, where τ is the time constant for which the amplitude of the vibration decays to $1/e$ of its initial value.

Modeling the frequency-dependent losses is again a more demanding task. By applying the central difference scheme to the third-order temporal derivative of Eq. 2.7, we have [10]

$$y_{ttt}(t,x) \approx y(n+2,m) - 2y(n+1,m) + 2y(n-1,m) - y(n-2,m), \tag{4.24}$$

where the term $y(n+2,m)$ would require information about the upcoming state of the string, which is not known at time instant $n+1$ when Eq. 4.24 is evaluated, and the scheme thus becomes implicit. Since the frequency-dependent damping term is relatively small, it is possible

to approximate it using Eq. 4.11 so that the number of temporal steps will not be increased and the number of spatial steps will be restricted to four. This somewhat complex recurrence equation can be found in [10] and is not repeated here.

Equation 4.21 is actually a special case of a more general system, where both terms $y(n, m+1)$ and $y(n, m-1)$ have different coefficients, namely g_m^+ and g_m^-. This structure is called a 1D *finite difference time-domain* (FDTD) waveguide [32], and it is illustrated in Figure 4.3. This formulation can be used naturally for modeling the frequency-independent losses, but it

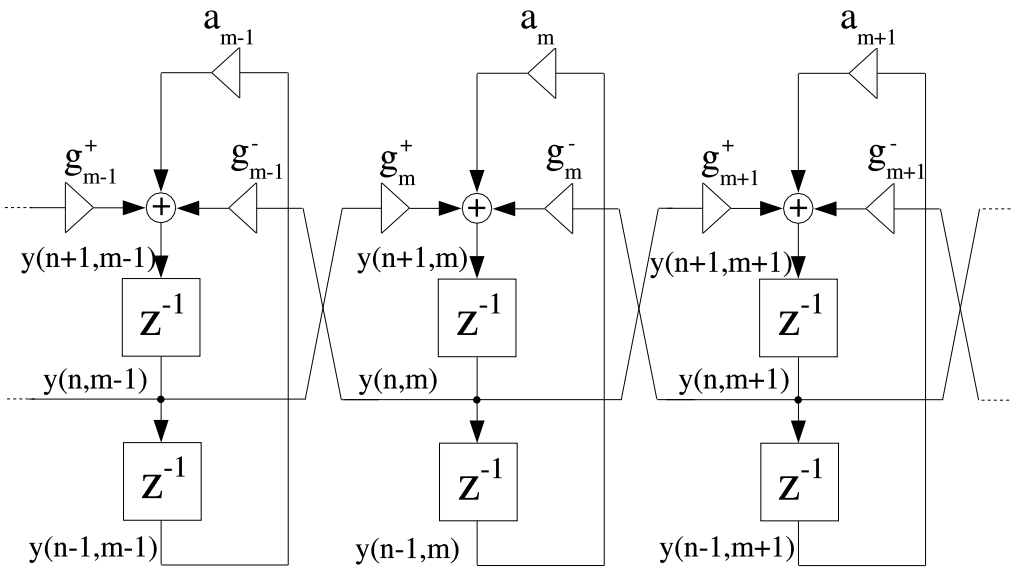

Figure 4.3: Digital waveguide implementation of the 1D wave equation's finite difference approximation (after [32]).

is also cabable of simulating *scattering junctions*, a generalization of terminations, if g_m and a_m values are chosen correctly [32]. Scattering junctions are used in simulating impedance discontinuities, e.g. when the cross-section area of an acoustic tube changes, or when a string is connected to the soundboard via a nonrigid termination. Since we are using a simplified model for the string termination, and since the mechanical impedances in strings are generally uniform, we will not go to further details here. Also, since the losses generally take place uniformly throughout the string (excluding the terminations), we could give up the dependency of g_m and a_m from the spatial location m. We will, however, leave the subscripts intact in order to better distinguish the FDTD loss parameters from the DWG loss parameter g, and allpass-coefficient a.

The FDTD waveguide can also be used in modeling fractionally positioned string terminations, thus giving an alternative tool for tuning the string model. If the g_m and a_m parameters are chosen so that [32]

$$g_m^+ \quad = \quad 1 - \delta$$

$$\begin{aligned} g_m^- &= 1+\delta \\ a_m &= -2.0, \end{aligned}$$

then the effective reflection point at low frequencies is at $m+\delta$. This operation corresponds to first-order Lagrange interpolation, a fractional delay technique not unlike we encountered in Section 3.5.1. It must be noted, however, that since the location of the termination depends on the frequency, this technique is applicable only with high sampling rates or with systems that have heavy damping at high frequencies [32].

Finite difference models generally have stability problems if the spatial sampling frequency is not sufficiently high. This is due to the grid dispersion phenomenon discussed in Section 4.2.1. Instability occurs when the string vibration starts to increase without addition of external excitation, and therefore stability can be assured by applying suitable damping on the system. It has been proven in [18], that choosing

$$|g_m^+| = |g_m^-| = |g_m| \leqslant 1 \ , \ a_m = -g_m^2, \tag{4.25}$$

the stability of an LTI 1D FDTD string model can be guaranteed. In addition to this, the simple form of Eq. 4.25 and the fact that we are not using physical quantities (such as d_1 in Eqs. 4.22 and 4.23) in modeling the string losses, suggests us to use this approach in our nonlinear FDTD string model.

4.3.1 Stiffness in finite difference strings

The dispersive behavior of strings is best simulated in FDM models by directly discretizing the stiff 1D wave equation (Eq. 2.8) using the same finite difference scheme as before (Eq. 4.2) [27]. This leads to a recurrence equation containing three additional spatio-temporal terms in the discretized wave equation. The recurrence equation for stiff strings has been covered in earlier studies (e.g. [26], [10]) and will not be repeated here.

4.4 Nonlinearities in Finite Difference Models

As suggested in [32], the hard-limiting effects discussed in Section 2.2.4 can be modeled by inserting an impulse excitation at the location of the limiter. The procedure is the same as in the case of a DWG model, discussed in Section 2.2.4, because the string model used in [32] is actually an FDTD waveguide. Nonlinear interactions between an excitor and an FDM string, such as the hammer-string and the bow-string interactions, have been discussed in [9], [11], and [12], and in [26], [49], and [50], respectively.

4.4.1 Implementing TM in an FDTD string model

Implementing tension modulation in a digital waveguide string in Section 3.5 was not an overly difficult task. This was due to the fact that the implementation of a DWG string is essentially a feedback loop with delay, and therefore modulating the delay time of this loop corresponded to modulating the wave velocities. In FDM strings, however, such an approach would not lead to satisfactory results, since the physical quantities (e.g. displacement or velocity) themselves are present in the string model, instead of their wave decompositions[2].

One approach could be to substitute fractional delay elements for the unit delays in the 1D FDTD model, illustrated in Figure 4.3. This would, however, form a delay-free loop, and thus lead to an implicit equation. Inserting unit delays between the fractional delays to avoid the delay-free loops would not yield succesful results either, because that would violate the recurrence equation 4.11, which states that the spatial and temporal distances between the terms of Eq. 4.11 must remain the same. On a spatio-temporal grid this means that the spatial resolution must equal the temporal resolution. Varying the tuning coefficient r would, in turn, likely result in problems with the grid dispersion.

Due to these restrictions we concluded that in order for the system to work properly, we would first have to evaluate the recurrence equation, interpolate the elongated string shape from that, and then re-evaluate the recurrence equation using the interpolated values. This can also be seen as using two FDTD systems in implementing the nonlinear string. The elongation of the string would be evaluated from one system, and the result, the stretched string state, would be updated to the other system. Figure 4.4 illustrates this procedure on the spatio-temporal grid.

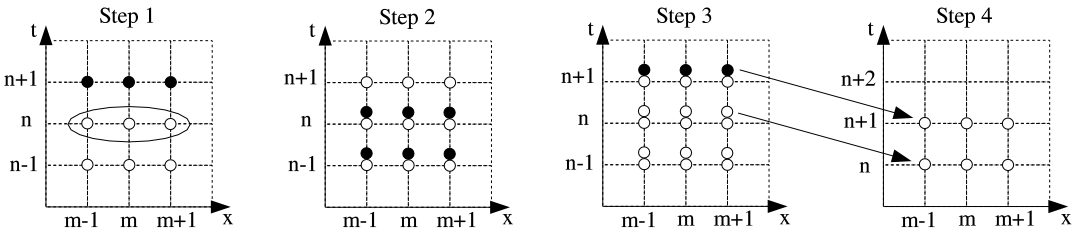

Figure 4.4: Illustration of the nonlinear FD algorithm on a spatio-temporal grid. The vertical axis denotes the time, while the horizontal axis denotes the spatial location on the string. The illustration is shown only for a string segment of length $L_{nom} = 3$, for clarity. In each step of the algorithm, most recently evaluated values are presented as black dots, while earlier values are presented as white dots.

In step 1, the two initial states have been assigned for the string, and the state at the next

[2]Such a system which deals with the physical quantities themselves is called a *Kirchhoff* model, as opposed to a *wave* model, which deals with the wave components of the physical quantities.

instant (in the linear case) is obtained by the standard recurrence equation (Eq. 4.11). The grid values which represent the state of the string at the corresponding time instant are circled in step 1. In step 2, sample values corresponding to the TM have been interpolated from the string states in step 1. In step 3, Eq. 4.11 has been applied on the values evaluated in step 2, in order to obtain the string state corresponding to the change in tension. The two most recently obtained states are now taken as the "initial states" in step 4, and we can return to step 1. As seen in Fig. 4.4, the tension modulation corresponds here to interpolating the string state in the temporal domain. The elongation of the FDTD string was evaluated similarly to what was done in Equation 3.7, except that here the slope of the string was obtained by taking the difference of the displacements between two adjacent string segments, rather than summing up the slope wave components. In the following, we will have a closer look at the interpolation process.

4.4.2 String state interpolation

We chose again to use first-order allpass filters in interpolating the string state from the linear model (Step 2 in Fig. 4.4). Figure 4.5(a) illustrates how the interpolated value of $y(n, m)$ is obtained from the linear values. The spatio-temporal grid on the left represents the string state in the linear case, while the spatio-temporal grid on the right represents the string state after spatial interpolation. The structure between the two grids is the block diagram of a first-order allpass filter (Eq. 3.4). The coefficient a for the allpass filter was evaluated as presented earlier by Eqs. 3.7-3.9.

From this figure, we notice that the allpass filter uses the value of $y(n + 1, m)$ delayed by one time sample, thus corresponding to $y(n, m)$. Clearly, this can be obtained directly from the grid on the left, and the branch on the left containing the unit delay can be reformed. The result is shown in Figure 4.5(b). Here we also note that the interpolation system uses its own output at the previous time instant. This is actually the same as using the value of $y(n - 1, m)$, because it is the same as the output of the interpolation process one time step ago (this might be best understood by noting that the bottom row of step 4 in Fig. 4.4 is the same as the bottom row of step 1 at the next time instant). Thus, Fig. 4.5(b) can be further simplified to Fig. 4.5(c).

Having this said, the recurrence equation for the nonlinear finite difference string can be written as

$$\begin{aligned}
y_1(n+1,m) &= g_m(y_2(n,m-1) + y_2(n,m+1)) - a_m y_2(n-1,m),\\
y_2(n,m-1) &= -a y_1(n+1,m-1) + y_1(n,m-1) + a y_1(n-1,m-1),\\
y_2(n,m+1) &= -a y_1(n+1,m+1) + y_1(n,m+1) + a y_1(n-1,m+1),\\
y_2(n-1,m) &= -a y_1(n,m) + y_1(n-1,m) + a y_1(n-2,m).
\end{aligned} \tag{4.26}$$

Here the subscript indices 1 and 2 refer to the linear and interpolated strings, respectively. Simplifying and rearranging we end up with an equation containing only terms of y_1, and the

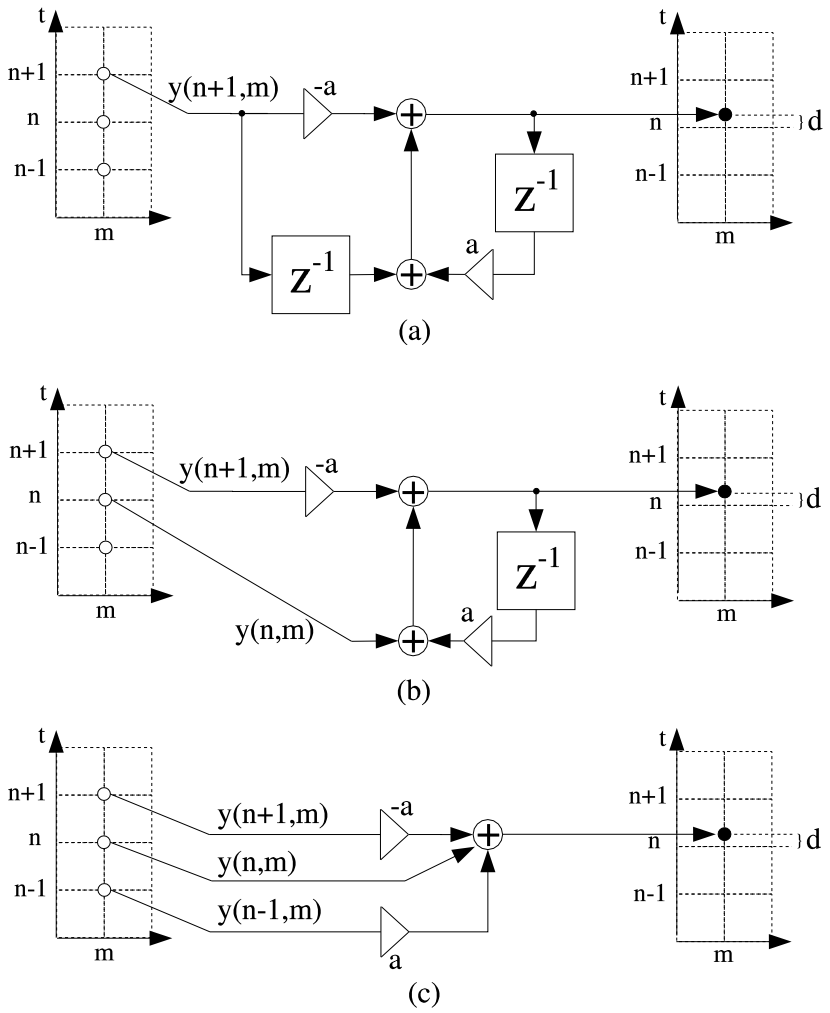

Figure 4.5: Illustration of the interpolation process due to the change in the string's length. The spatio-temporal grids on the left and right represent the linear and interpolated string states, respectively. The fractional delay value caused by the interpolation is denoted by d. The interpolation process in (a) is simplified in (b), and further in (c).

subscript may therefore be omitted

$$\begin{aligned} y(n+1,m) &= -g_m a y(n+1,m-1) - g_m a y(n+1,m+1) + g_m y(n,m+1) \\ &+a_m a y(n,m) + g_m y(n,m-1) + g_m a y(n-1,m+1) - a_m y(n-1,m) \\ &+g_m a y(n-1,m-1) - a_m a y(n-2,m) \end{aligned} \tag{4.27}$$

This equation is illustrated with a block diagram in Figure 4.6 along with its abstraction. A nonlinear FDTD string can be constructed by connecting several of these blocks together. We will refer to such a block as a *nonlinear FDTD (NFDTD) element*. Illustration of the lossless NFDTD element can be found in Figure 4.7, where g_m and a_m equal unity and have therefore

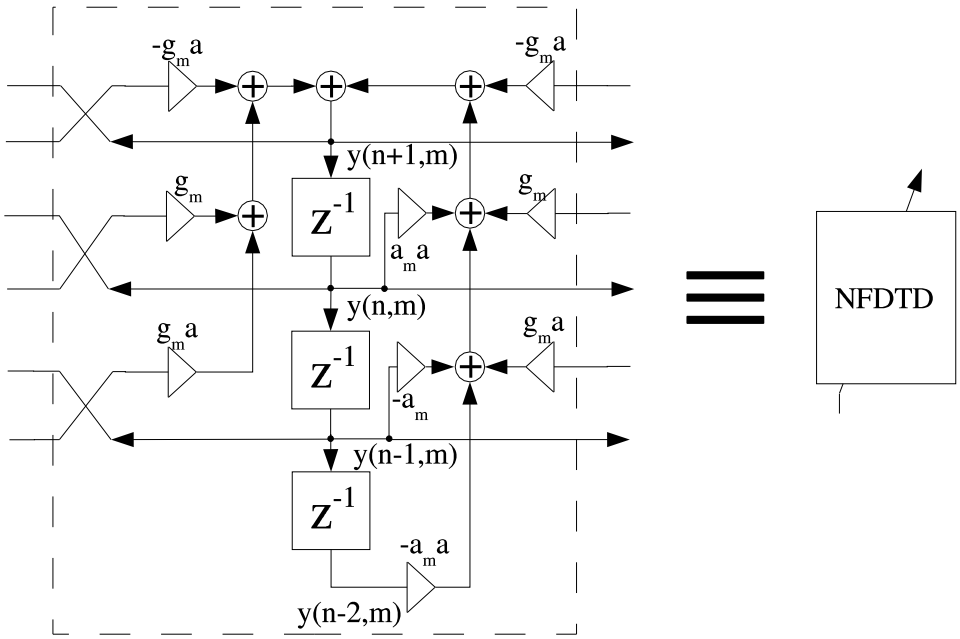

Figure 4.6: Illustration of the lossy nonlinear FDTD element.

been left out.

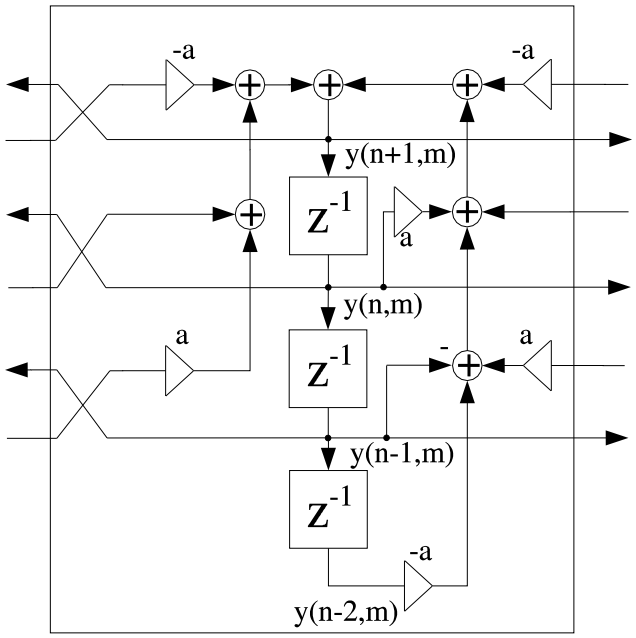

Figure 4.7: Illustration of the lossless nonlinear FDTD element.

4.4.3 String excitation and termination

For the interaction with the FDTD string model, we chose to use the "boxcar" excitation model discussed in Section 4.2.2, so that the excitation signal could again be interpreted as a force signal. Figure 4.8 presents an interaction block to be used with a nonlinear FDTD string. We will call such a block the *FDTD interaction element*.

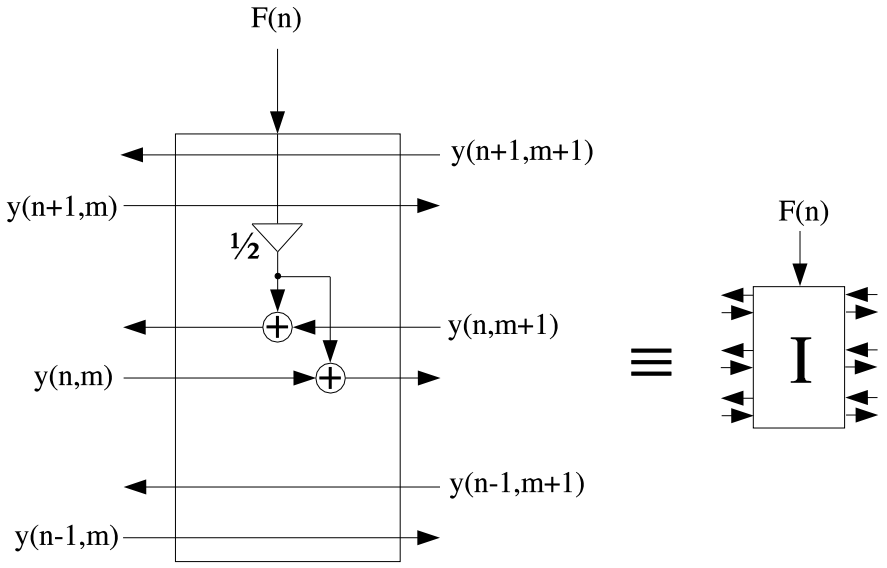

Figure 4.8: Illustration of the FDTD interaction element together with its abstraction. The excitation algorithm is defined by Eqs. 4.18 and 4.19.

Using these DSP blocks, we can construct a one-polarization NFDTD string, as illustrated in Figure 3.8. We chose to use rigid terminations for our nonlinear finite difference string model, since the modeling of frequency-dependent losses at the terminations is not a key aspect of this study. Fixed terminations do not ruin the generation of missing harmonics in our model either, since the TMDF coupling is implemented in a different manner, as we are about to see in the following section. The zero-blocks, introduced in Fig. 4.9, are elements that merely give zeros to all outputs for all input signal values.

4.4.4 Simulation results and comparison to measured data

As suspected, the nonlinear FDTD algorithm is cabable of simulating the fundamental frequency glide phenomenon of real musical instruments. Figure 4.10 depicts the fundamental frequencies of a measured kantele tone and a nonlinear FDTD string tone as a function of time. The measured kantele tone is the same one used in Fig. 3.13. The string model used here consists of 56 NFDTD elements, and the tuning coefficient (Eq. 4.8) has a value of $r = 0.998$. The allpass coefficient a is scaled using a coefficient 0.25 in order to correctly simulate the

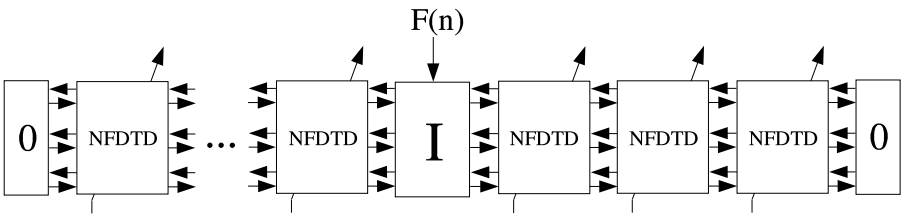

Figure 4.9: One-polarizational nonlinear FDTD string. The string consists of the nonlinear FDTD elements illustrated in Fig. 4.6. The zero-blocks at the terminations give zero as an output regardless of the input values, thus implying rigid terminations. The excitation to the string can be inserted as a force signal using an FDTD interaction element, illustrated in Fig. 4.8.

recorded tone.

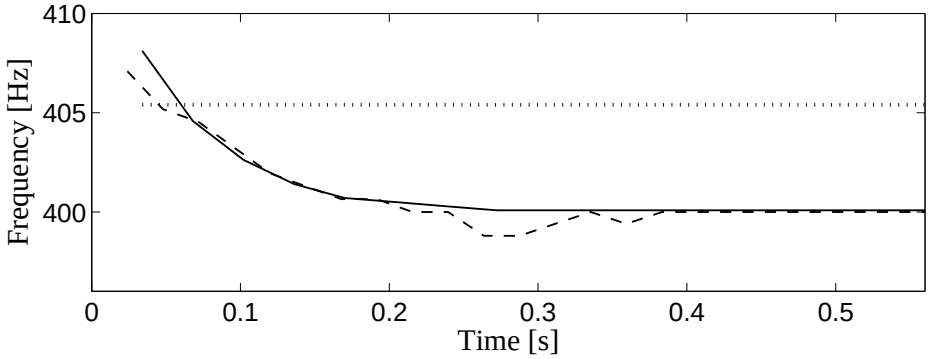

Figure 4.10: Illustration of the fundamental frequency glide in the nonlinear FDTD tone (solid line) and in a real kantele tone, obtained via measurements (dashed line). The dotted horizontal line presents, as before, the approximated detection threshold of a pitch drift.

When modeling the generation of missing harmonics in the nonlinear DWG string, we exploited the computational error in the elongation calculation, and were able to turn the error to our advantage. Modeling the generation of missing harmonics in a nonlinear FDTD string is however not so simple. Even if a leaky integrator is used in the elongation calculation, its parameters do not have a desirable effect on the missing harmonics. This effect does not seem too surprising when considering the major differences of these two algorithms. Instead, we constructed a model, where an additional interaction element is placed between the last NFDTD element and the string termination. After evaluating the TMDF, we can insert it into the additional interaction element, because the interaction element is able to excite the string during run-time, and it takes a force signal as its input.

The vertical component of the TMDF can be seen to be directly proportional to the product of the displacement of the last NFDTD element and the tension. Figure 4.11 clarifies this. Also,

since the tension variation is directly proportional to the elongation signal, we can obtain the TMDF by multiplying the elongation signal by the displacement of the last NFDTD element, and scaling the product with a constant coefficient. This coefficient, denoted α_{TMDF}, can now be used to control the amplitude of the missing harmonics. A nonlinear FDTD string model with the generation of missing harmonics is presented in Figure 4.12.

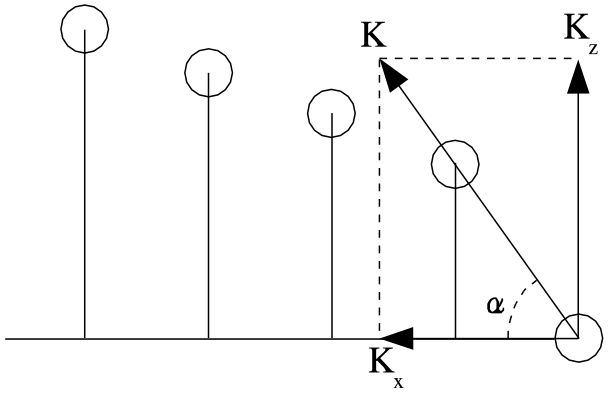

Figure 4.11: Illustration of the NFDTD string termination. The arrows denote the tension modulation driving force, while the circles denote the displacements of different NFDTD elements. The vertical component of the tension is directly proportional to the displacement of the last NFDTD element, since $tan(\alpha) = K_z/K_x = y(N, m_L - 1)/1$, where m_L is the index of the last NFDTD element. Clearly also $K_z = y(N, m_L - 1)K_x$.

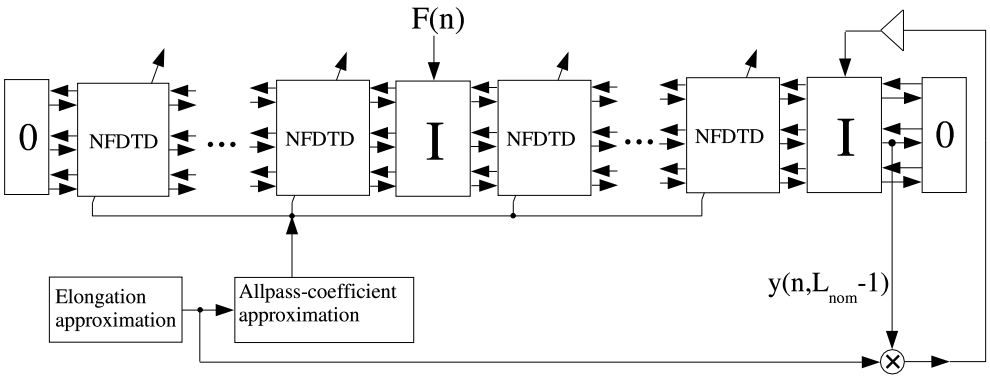

Figure 4.12: Illustration of the NFDTD string with a generation mechanism for missing harmonics. A second interaction element is added in order to feed the TMDF into the string. The scaling coefficient α_{TMDF} controls the amplitude of the missing harmonics. The string elongation is approximated from the displacements of each NFDTD element.

Figure 4.13 represents the behavior of the first three harmonics of a tone synthesized by this model. It can be seen that the missing harmonics can be “lifted” by choosing a proper

value for α_{TMDF}. The stability of the system, however, poses an upper limit for the α_{TMDF} coefficient, since the TMDF mechanism continuously feeds energy to the string. According to our experience, missing harmonics with amplitudes greater than what shown in Fig. 4.13 is difficult.

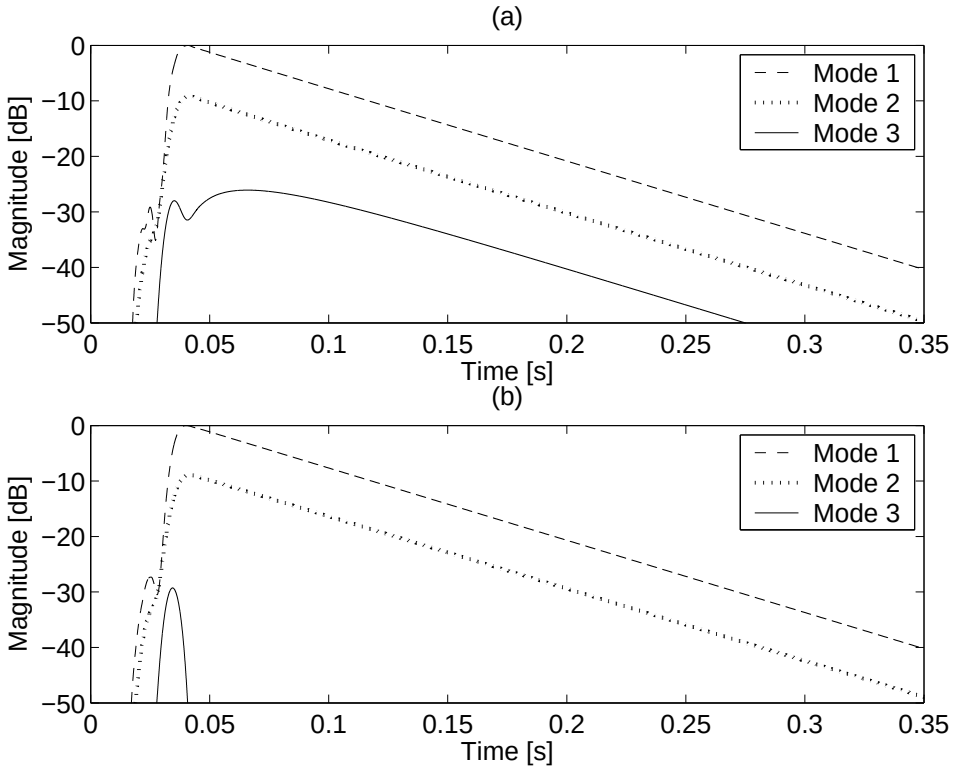

Figure 4.13: Generation of missing harmonics in a nonlinear FDTD string. The string was plucked again approximately at $1/3$rd of its length, and the coupling of the TMDF to the transversal vibration was controlled using a scaling coefficient α_{TMDF}. In (a), the scaling coefficient has a value of $\alpha_{TMDF} = 4$, and the missing third harmonic can be seen rising after the initial transient. In (b), $\alpha_{TMDF} = 0$, and generation of missing harmonics doesn't take place.

4.5 Stability Analysis of the Nonlinear FDTD Algorithm

We studied the stability of the nonlinear FDTD algorithm using the Von Neumann analysis [58] in the time-invariant case, i.e. parameter a of Eq. 4.27 was kept constant. The basic idea of this method is to calculate the spatial Fourier spectrum of the system under discussion at two consecutive time steps. An amplification function, which shows how the spatial spectrum evolves with time, can then be derived from the two spectra. If the absolute value of this

amplification function remains below unity, stability is guaranteed. Since the Von Neumann analysis of complicated recurrence equations is not necessarily self-evident, we will discuss this a bit more thoroughly here.

If the spatial inverse Fourier transform is defined

$$y(n,m) = \mathscr{F}\{Y(n,\xi)\}^{-1} = \gamma^n e^{im\theta}, \tag{4.28}$$

the nonlinear finite difference recurrence equation 4.27 can be written as

$$\begin{aligned}&\gamma^{n+1}e^{im\theta} = -g_m a\gamma^{n+1}e^{i(m+1)\theta} - g_m a\gamma^{n+1}e^{i(m-1)\theta} + g_m\gamma^n e^{i(m+1)\theta}\\ &+a_m a\gamma^n e^{im\theta} + g_m\gamma^n e^{i(m-1)\theta} + g_m a\gamma^{n-1}e^{i(m+1)\theta} - a_m\gamma^{n-1}e^{im\theta}\\ &+g_m a\gamma^{n-1}e^{i(m-1)\theta} - a_m a\gamma^{n-2}e^{im\theta}.\end{aligned} \tag{4.29}$$

Dividing with $\gamma^{n-2}e^{im\theta}$ and rearranging, we have

$$\begin{aligned}&(1 + g_m a e^{i\theta} + g_m a e^{-i\theta})\gamma^3 + (-g_m e^{i\theta} - a_m a - g_m e^{-i\theta})\gamma^2\\ &+(-g_m a e^{i\theta} + a_m - g_m a e^{-i\theta})\gamma - a_m a = 0.\end{aligned} \tag{4.30}$$

Using the Euler's equation leads to a simpler form

$$A\gamma^3 + B\gamma^2 + C\gamma + D = 0, \tag{4.31}$$

where

$$\begin{aligned}A &= 1 + 2g_m a cos(\theta)\\ B &= -a_m a - 2g_m cos(\theta)\\ C &= a_m - 2g_m a cos(\theta)\\ D &= -a_m a.\end{aligned} \tag{4.32}$$

In order to get the amplification function γ, we would now have to solve the third-order equation 4.31. Unfortunately, the solution of this equation is complicated and involves dozens of terms. If we want to consider the stability of the lossless nonlinear FDTD string, we can substitute $g_m = a_m = 1$. This simplifies the solution of Eq. 4.31 enough to enable numerical stability analysis for the amplification function. The absolute value of the amplification function γ is illustrated in Figure 4.14 as a function of the interpolation coefficient a, and the spatial frequency θ.

It is important to note that this stability analysis is conducted on a *lossless* nonlinear FDTD string with *constant interpolation coefficient*. We can thus call this system time-invariant (normally the interpolation coefficient depends on the string elongation). Figure 4.14 reveals that in the lossless case, the time-invariant version of the nonlinear FDTD algorithm is unstable for all but very small a parameter values. Making the algorithm time-variant results in an even less stable system. In a practical lossy string implementation, however, the nonlinear FDTD string remained stable for typical excitation amplitudes (i.e. excitation amplitudes commonly used when playing real string instruments).

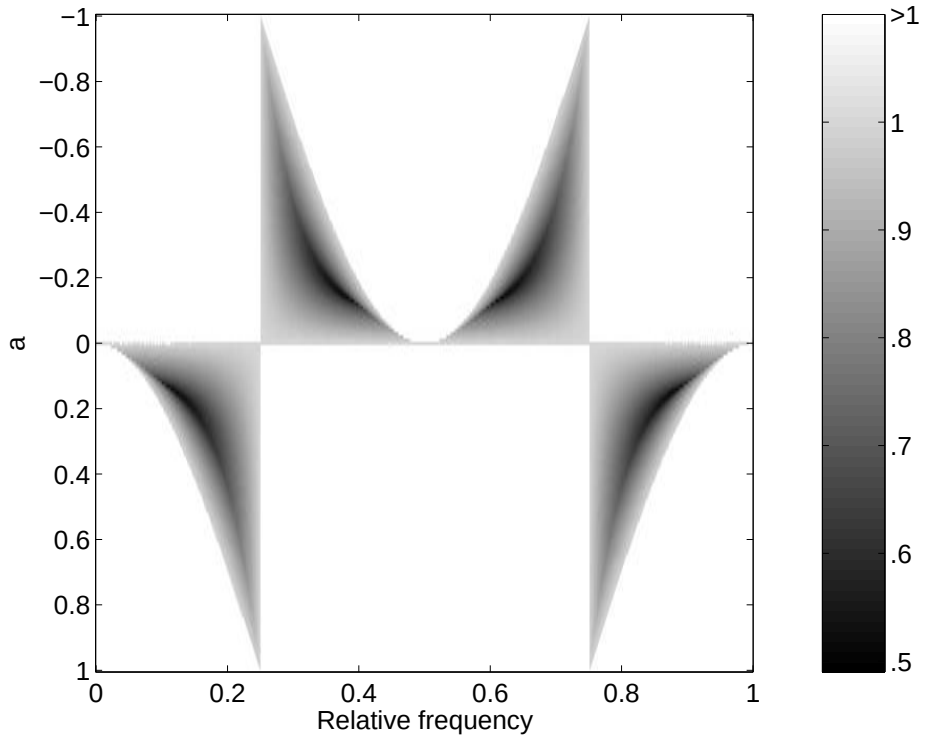

Figure 4.14: Absolute value of amplification function γ of a nonlinear FDTD algorithm as a function of the allpass-coefficient a and the relative frequency θ. The white color denotes areas where the amplification function exceeds unity, i.e. when the model becomes unstable.

4.6 Comparison of the Nonlinear Algorithms

Now that both of the spatially distributed nonlinear string algorithms have been thoroughly discussed, it is probably a good idea to have a brief comparison between the two models. At first glance the algorithms seem quite alike; both models are spatially distributed, and the user interface for both algorithms is similar. The excitation for both models is inserted during runtime as a force signal into an interaction element and the amount of nonlinearity and the direct coupling of the TM can be controlled via scaling coefficients. Also, the TM signal computation is exactly the same in both models, and the initial pitch glide can be modeled quite similarly using both models, unless heavily exaggerated pitch glides are desired.

A more careful examination, however, reveals also many differences. The inner structures of the models are not at all similar; the signals inside the nonlinear DWG model can be interpreted as wave components (velocity waves, in this case), while the signals inside the nonlinear FDTD model present the actual physical quantities (displacement of the string). This is the fundamental distinction between wave- and Kirchhoff models, and it is present in all DWG and finite difference models. Also, due to the loop-like structure of the nonlinear DWG model, im-

plementing frequency-dependent losses can be done simply by inserting a loop filter, whereas the nonlinear FDTD model would require adding extra terms in the original wave equation for realistic loss modeling.

The computational complexities of the algorithms are also different. Since the models consist mainly of the basic string blocks (basic elements in the DWG case and NFDTD elements in the finite difference case), the differences in the computation of the basic string blocks dominates the computational needs of the algorithms. The basic element (Fig. 3.7) consists of four multiplications and two summations per time sample, whereas the nonlinear FDTD element (Fig. 4.6) requires a total of nine multiplications and eight summations for computing one time sample. Although the interaction and termination blocks are much simpler in the finite difference case, the typically large number of the string elements turns the favor to the nonlinear DWG model. The cost for the allpass-coefficient estimations and memory requirements for the two models are approximately the same. Figure 4.15 depicts the number of total computational operations per time sample as a function of string elements for both of the two models.

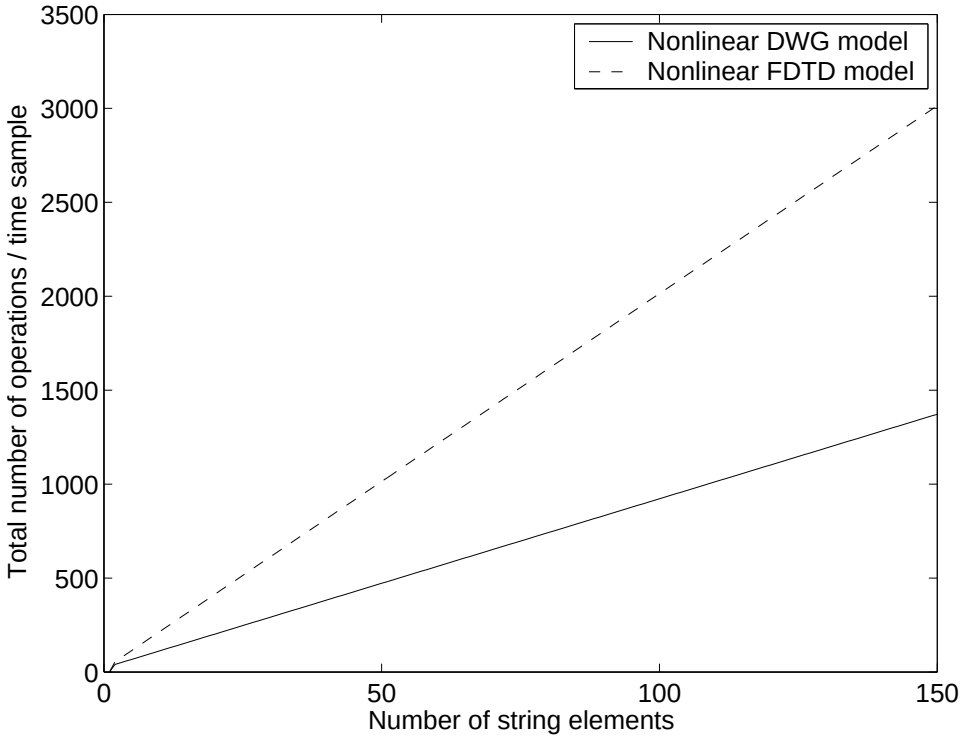

Figure 4.15: Number of computational operations required for each time sample as a function of the number of string elements. The computational requirements increase linearly in both cases, although the slope is greater for the nonlinear FDTD string algorithm.

In the figure we can see, e.g. that if a nonlinear string with a nominal fundamental frequency of 440 Hz and a temporal sampling frequency of 44.1 kHz is desired, the nonlinear FDTD

algorithm would consist of 50 elements, and would therefore need a computational power of 45 MFLOPS (millions of floating point operations per second), while the nonlinear DWG algorithm would settle for 22 MFLOPS. From this approximation, we can conclude that in theory, a modern PC with a computational power of 3500 MFLOPS would be able to run over 150 nonlinear DWG string models, or over 70 nonlinear FDTD string models in real-time. In practice, however, the operating system, other processes, and implementational issues slow the performance down considerably.

Besides the computational complexities, another important difference between the two presented models is stability. As discussed in Sections 3.5.3 and 4.5, the nonlinear FDTD algorithm has far more severe stability problems than its digital waveguide counterpart. Both models perform equally well when modeling the initial pitch glide phenomenon of existing musical instruments, but exaggerating the glides results into stability problems much sooner with the NFDTD model than with the NDWG one. Also, as discussed in section 4.4.4, the stability of the NFDTD algorithm limits the obtainable amplitudes of the missing harmonics, so that realistic modeling of this phenomenon becomes difficult. Fortunately, the nonlinear DWG algorithm does not have this constriction.

Chapter 5

Conclusions and Future Work

This chapter will briefly summarize the main results of this thesis and discuss where these models could be used. Also, suggestions for possible future studies are presented.

5.1 Main Results of This Thesis

Two algorithms for modeling spatially distributed nonlinear strings in a physically meaningful way were presented. The nonlinear digital waveguide model uses first-order allpass filters in modulating the total delay time of the waveguide, which corresponds to a modulation of the fundamental frequency in the resulting tone. The spatially distributed nature of this model is the real novelty here, and it enables interaction with the string at any spatial location during runtime. The nonlinear digital waveguide algorithm is able to closely simulate the fundamental frequency glide phenomenon of real strings, as well as considerably exaggerate it. Also, the generation of missing harmonics in real strings can be reasonably well emulated using the digital waveguide model, although the computational process producing it is not physically valid.

The second algorithm, a spatially distributed finite difference model of a nonlinear vibrating string, uses first-order allpass filters in interpolating the string state between consecutive time samples in the lossy finite difference recurrence equation. This is essentially the same as modulating the procession rate of the recurrence equation. Finite difference methods have not been used in modeling nonlinear strings up until a very recent study by Bilbao [8]. Our modeling approach, however, is considerably different. The nonlinear finite difference algorithm presented in this thesis is capable of realistically simulating the fundamental frequency glide, as well as exaggerate it to some extent. The generation of missing harmonics is modeled by this algorithm in a pseudo-physical manner, although stability problems make the realistic simulation of this phenomenon difficult. Comparison of the two methods reveals that the nonlinear finite difference string algorithm is computationally twice as expensive as the digital

waveguide version. The main results of this thesis will published later as a journal article [44].

5.2 Application Possibilities and Future Work

A natural application area for these string models is the electronic music industry. If combined with proper instrument body models and reverberation algorithms, the string models could be used in hardware or software synthesizers, resulting in high quality string instrument tones with a great degree of flexibility. Also, the gaming industry and virtual reality applications could gain from the string models. As an example of the latter, the nonlinear digital waveguide algorithm was used in the implementation of a *virtual kantele*, in the experimental virtual environment (EVE) at the Telecommunications Software and Multimedia Laboratory, Helsinki University of Technology. Figure 5.1 illustrates the graphical user interface of the virtual kantele. For more information about the EVE, see [1].

In the virtual kantele demo, five virtual strings are projected in the EVE-room, and the user is able to excite them using magnetically-tracked data gloves. An illusion of a 3D environment is generated via magnetically tracked stereo shutter glasses, synchronized to a set of projectors. When the user excites a string, the resulting sound is generated via loudspeakers using the kantele string model, discussed in Sec. 3.6.2, running on a software platform called Mustajuuri [29]. The plucking velocity on the virtual string defines the amplitude of the excitation, while the plucking location defines the amount of nonlinearity in the sound. The user can also generate tones with negative nonlinearities, i.e. ascending pitch glides.

The formulation of energetically valid nonlinear string models (i.e. models where the string's energy behaves in a physically correct way), as well as embedding the nonlinear string algorithms in a full-scale physical model of a plucked-string instrument body, are left for future work.

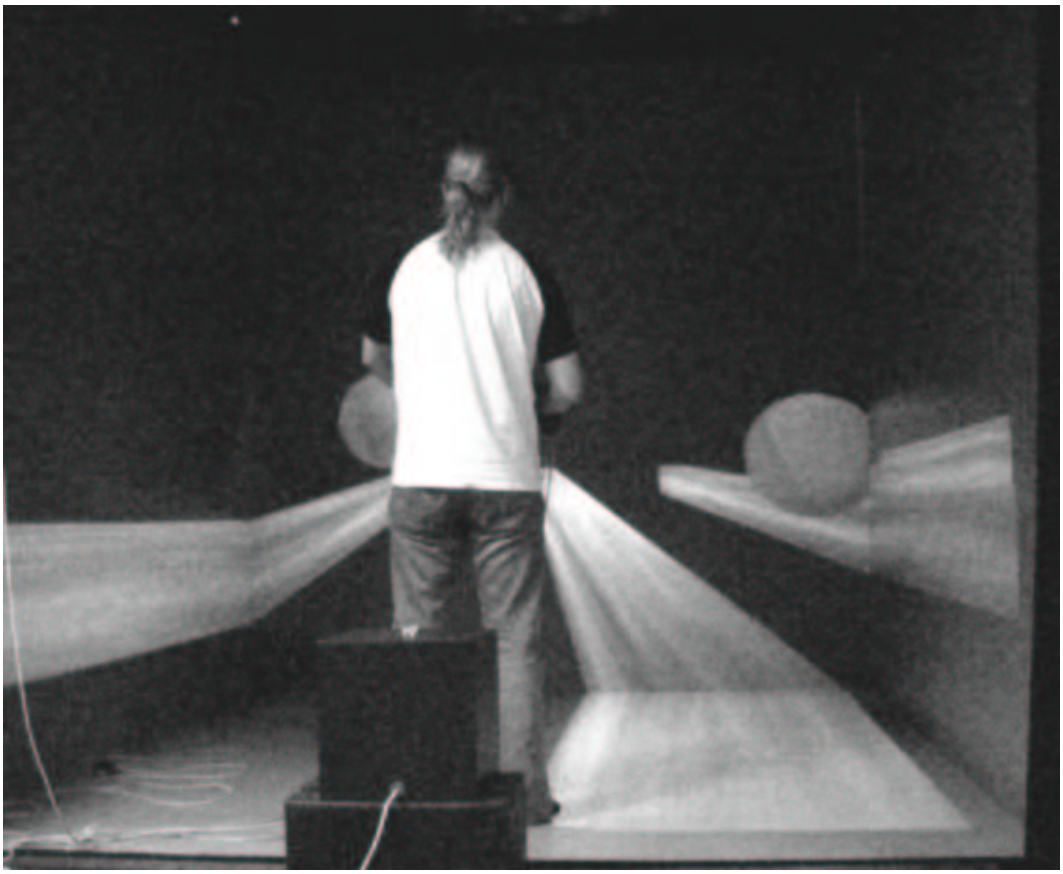

Figure 5.1: Illustration of the virtual kantele at EVE. The user can excite the virtual strings (projected bars) using data gloves, and the resulting sound is played through a set of loudspeakers. The velocity of the excitation defines the resulting amplitude of the vibration, and the excitation location defines the amount of nonlinearity. The projected spheres denote the locations of the user's hands in the 3D environment.

References

[1] EVE - the experimental virtual environment. http://eve.hut.fi/, 2001. Referred May 6, 2004.

[2] J.-M. Adrien. The missing link: Modal synthesis. *G. De Poli, A.Piccialli and C. Roads eds. Representations of Musical Signals. MIT Press, Cambridge, Massachusetts*, pages 269–297, 1991.

[3] B. Bank. Nonlinear interaction in the digital waveguide with the application to piano sound synthesis. In *Proceedings of the International Computer Music Conference*, pages 54–58, Berlin, Germany, September 2000.

[4] B. Bank and L. Sujbert. Modeling the longitudinal vibration of piano strings. In *Proc. Stockholm Music Acoustics Conference*, pages 143–146, Stockholm, Sweden, August 6-9, 2003.

[5] J. Bensa, S. Bilbao, R. Kronland-Martinet, and J. O. Smith. A power normalized nonlinear lossy piano hammer. In *Proc. Stockholm Music Acoustics Conference*, volume I, pages 365–368, Stockholm, Sweden, August 6-9, 2003.

[6] J. Bensa, K. Jensen, R. Kronland-Martinet, and S. Ystad. Perceptual and analytical analysis of the effect of the hammer impact on the piano tones. In *Proceedings of the International Computer Music Conference*, pages 58–61, Berlin, Germany, September 2000.

[7] S. Bilbao. *Wave and Scattering Methods for the Numerical Integration of Partial Differential Equations*. PhD thesis, Stanford University. Available on-line at http://ccrma-www.stanford.edu/∼bilbao/, 2001.

[8] S. Bilbao. Energy-conserving finite difference schemes for tension-modulated strings. In *the International Conference on Acoustics, Speech and Signal Processing*, Montreal, Canada, May 17-21, 2004.

[9] X. Boutillon. Model for piano hammers: Experimental determination and digital simulation. *Journal of the Acoustical Society of America*, 83:746–754, 1988.

[10] A. Chaigne. On the use of finite differences for musical synthesis. Application to plucked stringed instruments. *Journal d'Acoustique*, 5:181–211, 1992.

[11] A. Chaigne and A. Askenfelt. Numerical simulations of piano strings. I. A physical model for a struck string using finite difference methods. *Journal of the Acoustical Society of America*, 95(2):1112–1118, February 1994.

[12] A. Chaigne and A. Askenfelt. Numerical simulations of piano strings. II. Comparisons with measurements and systematic exploration of some hammer-string parameters. *Journal of the Acoustical Society of America*, 95(3):1631–1640, March 1994.

[13] A. Chaigne and C. Lambourg. Time-domain simulation of damped impacted plates. Part I. Theory and experiments. *Journal of the Acoustical Society of America*, 109(4):1422–1432, April 2001.

[14] J. M. Chowning. The synthesis of complex audio spectra by means of frequency modulation. *Journal of the Acoustical Society of America*, 21(7):526–534, 1973.

[15] H. A. Conklin. Generation of partials due to nonlinear mixing in a stringed instrument. *Journal of the Acoustical Society of America*, 105(1):536–545, January 1999.

[16] L. Cremer. *The Physics of the Violin*. MIT Press, Cambridge, MA, 1983.

[17] W. C. Elmore and M. A. Heald. *Physics of Waves*. Dover Publications, Inc., 1985.

[18] C. Erkut and M. Karjalainen. Virtual strings based on a 1-D FDTD waveguide model: Stability, losses, and traveling waves. In *Proceedings of the Audio Engineering Society 22nd International Conference*, pages 317–323, Espoo, Finland, June 15-17, 2002.

[19] C. Erkut, M. Karjalainen, P. Huang, and V. Välimäki. Acoustical analysis and model-based sound synthesis of the kantele. *Journal of the Acoustical Society of America*, 112:1681–1691, October 2002.

[20] A. Fettweis. Wave digital filters: Theory and practice. *Proceedings of the IEEE*, 74(2):270–327, February 1986.

[21] N. H. Fletcher and T. D. Rossing. *The Physics of Musical Instruments*. Springer-Verlag, New York, USA, 1988.

[22] J.-L. Florens and C. Cadoz. The physical model: Modeling and simulating the instrument universe. In G. De Poli, A. Piccialli, and C. Roads, editors, *Representations of Musical Signals*, pages 227–268, Cambridge, MA, 1991. MIT Press.

[23] D. Hall. Piano string excitation in the case of small hammer mass. *Journal of the Acoustical Society of America*, 79:141–147, 1986.

[24] D. Hall. Piano string excitation II: General solution for a hard narrow hammer. *Journal of the Acoustical Society of America*, 81:535–546, 1987.

[25] D. Hall. Piano string excitation III: General solution for a soft narrow hammer. *Journal of the Acoustical Society of America*, 81:547–555, 1987.

[26] L. Hiller and P. Ruiz. Synthesizing musical sounds by solving the wave equation for vibrating objects: Part I. *Journal of the Audio Engineering Society*, 19(6):462–470, June 1971.

[27] L. Hiller and P. Ruiz. Synthesizing musical sounds by solving the wave equation for vibrating objects: Part II. *Journal of the Audio Engineering Society*, 19(7):542–551, June 1971.

[28] J.-M. Holm. An FD finite width excitation modelling approach in violin sound synthesis. In *Proc. AES 22nd in Virtual, Synthetic and Entertainment Audio*, pages 256–261, Espoo, Finland, June 14-17, 2002.

[29] T. Ilmonen. Mustajuuri (Scorzonera hispanica) - audio application and toolkit. http://www.tml.hut.fi/∼tilmonen/mustajuuri/, 2001. Referred May 6, 2004.

[30] D. A. Jaffe and J. O. Smith. Extensions of the Karplus-Strong plucked string algorithm. *Computer Music Journal*, 7(2):56–69, summer 1983.

[31] H. Järveläinen and V. Välimäki. Audibility of initial pitch glides in string instrument sounds. In *Proceedings of the International Computer Music Conference*, pages 282–285, Havana, Cuba, 17-23 September 2001. Available on-line at http://lib.hut.fi/Diss/2003/isbn9512263149/article3.pdf.

[32] M. Karjalainen. 1-D digital waveguide modeling for improved sound synthesis. In *the International Conference on Acoustics, Speech and Signal Processing*, pages 1869–1872, Orlando, Florida, USA, May 13-17, 2002.

[33] M. Karjalainen. BlockCompiler: Efficient simulation of acoustic and audio systems. In *Proc. 114th AES Convention*, Amsterdam, The Netherlands, 22-25 March 2003.

[34] M. Karjalainen, V. Välimäki, and Z. Jánosy. Towards high-quality sound synthesis of the guitar and string instruments. In *Proceedings of the International Computer Music Conference*, pages 56–63, Tokyo, Japan, September 10-15, 1993.

[35] M. Karjalainen, V. Välimäki, and T. Tolonen. Plucked-string models: From the Karplus-Strong algorithm to digital waveguides and beyond. *Computer Music Journal*, 22(3):17–32, 1998.

[36] K. Karplus and A. Strong. Digital synthesis of plucked-string and drum timbres. *Computer Music Journal*, 7(2):43–55, 1983.

[37] T. I. Laakso, V. Välimäki, M. Karjalainen, and U. K. Laine. Splitting the unit delay - tools for fractional delay filter design. *IEEE Signal Processing Magazine*, 13(1):30–60, 1996.

[38] C. Lambourg, A. Chaigne, and D. Matignon. Time-domain simulation of damped impacted plates. Part II. Numerical model and results. *Journal of the Acoustical Society of America*, 109(4):1433–1447, April 2001.

[39] K. A. Legge and N. H. Fletcher. Nonlinear generation of missing modes on a vibrating string. *Journal of the Acoustical Society of America*, 76:5–12, July 1984.

[40] M. E. McIntyre and J. Woodhouse. On the fundamentals of bowed string dynamics. *Acustica*, 43(2):93–108, 1979.

[41] B.C.J. Moore. Frequency difference limens for short-duration tones. *Journal of the Acoustical Society of America*, 54:610–619, 1973.

[42] J. Pakarinen, M. Karjalainen, and V. Välimäki. Modeling and real-time synthesis of the kantele using distributed tension modulation. In *Proc. Stockholm Music Acoustics Conference*, volume 1, pages 409–412, Stockholm, Sweden, August 6-9, 2003.

[43] J. Pakarinen, M. Karjalainen, and V. Välimäki. Sound synthesis model for a nonlinear vibrating string using distributed fractional delay elements. In *Proc. Finnish Signal Processing Symposium*, pages 268–272, Tampere, Finland, May 19, 2003.

[44] J. Pakarinen, V. Välimäki, and M. Karjalainen. Physics-based methods for modeling nonlinear vibrating strings. *Acta Acustica united with Acustica*, 2004. Submitted for publication.

[45] M. Pavlidou and B. E. Richardson. The string-finger interaction in the classical guitar. In *Proc. Int. Symposium Musical Acoustics*, pages 559–564, 1995.

[46] M. Pavlidou and B. E. Richardson. The string-finger interaction in the classical guitar: Theoretical model and experiments. In *Proc. Int. Symposium Musical Acoustics*, pages 55–60, 1997.

[47] A. Pedersini, A. Sarti, and S. Tubaro. Object-based sound synthesis for virtual environments using musical acoustics. *IEEE Signal Process. Mag.*, 17(6):37–51, November 2000.

[48] J. R. Pierce and S. A. Van Duyne. A passive nonlinear digital filter design which facilitates physics-based sound synthesis of highly nonlinear musical instruments. *Journal of the Acoustical Society of America*, 101(2):1120–1126, February 1997.

[49] R. Pitteroff. Modelling of the bowed string taking into account the width of the bow. In *Proc. Stockholm Music Acoustics Conference*, pages 407–410, Stockholm, Sweden, 1993.

[50] R. Pitteroff and J. Woodhouse. Mechanics of the contact area between a violin bow and a string. Part I: Reflection and transmission behaviour. Part II: Simulating the bowed string. Part III: Parameter dependance. *Acustica-Acta Acustica*, pages 543–562, 1998.

[51] M. Podlesak and A. Lee. Dispersion of waves in piano strings. *Journal of the Acoustical Society of America*, 83(1):305–317, 1988.

[52] J. Pölkki, C. Erkut, H. Penttinen, M. Karjalainen, and V. Välimäki. New designs for the kantele with improved sound radiation. In *Proc. Stockholm Music Acoustics Conference*, pages 133–136, Stockholm, Sweden, August 6-9, 2003.

[53] C. Roads. *The Computer Music Tutorial*. The MIT Press, Cambridge, Massachusetts, USA, 1995.

[54] X. Rodet, Y. Potard, and J.-B. Barriére. The CHANT project: From the synthesis of the singing voice to synthesis in general. *Computer Music Journal*, 8(3):15–31, 1984.

[55] J. O. Smith. Physical modeling using digital waveguides. *Computer Music Journal*, 16(4):74–87, Winter 1992.

[56] J. O. Smith. Principles of digital waveguide models of musical instruments. *Applications of Digital Signal Processing to Audio and Acoustics, (M. Kahrs and K. Brandenburg, eds.)*, pages 417–466, February 1998.

[57] J. O. Smith. *Digital Waveguide Modeling of Musical Instruments*. http://www-ccrma.stanford.edu/ jos/waveguide/, [2004-03-17], May 9, 2002.

[58] J. C. Strikwerda. *Finite Difference Schemes and Partial Differential Equations*. Wadsworth, Brooks & Cole, California, USA, 1989.

[59] H. Suzuki. Model analysis of a hammer-string interaction. *Journal of the Acoustical Society of America*, 82:1145–1151, 1987.

[60] T. Tolonen, C. Erkut, V. Välimäki, and M. Karjalainen. Simulation of plucked strings exhibiting tension modulation driving force. In *Proceedings of the International Computer Music Conference*, pages 5–8, Beijing, China, October 22-28, 1999.

[61] T. Tolonen, V. Välimäki, and M. Karjalainen. Modeling of tension modulation nonlinearity in plucked strings. *IEEE Transactions on Speech and Audio Processing*, 8(3):300–310, May 2000.

[62] T. Tolonen, V. Välimäki, and M. Karjalainen. *Evaluation of Modern Sound Synthesis Methods*. Helsinki University of Technology, Acoustics Lab Report Series, Report no. 48. Available on-line at http://www.acoustics.hut.fi/publications/reports.html, 1998.

[63] L. Trautmann. *Digital Sound Synthesis by Physical Modeling of Musical Instruments using Functional Transformation Models*. PhD thesis, University of Erlangen-Nürnberg, 2002.

[64] L. Trautmann and R. Rabenstein. Digital sound synthesis based on transfer function models. In *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 83–86, New Paltz, NY, October 1999.

[65] V. Välimäki and T. I. Laakso. Principles of fractional delay filters. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 6, pages 3870–3873, Istanbul, Turkey, 5-9 June 2000.

[66] V. Välimäki, T. Tolonen, and M. Karjalainen. Plucked-string synthesis algorithms with tension modulation nonlinearity. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 2, pages 977–980, Phoenix, Arizona, March 15-19, 1999.

[67] V. Välimäki. *Discrete-Time Modeling of Acoustic Tubes Using Fractional Delay Filters*. Doctoral dissertation, Helsinki Univ. of Tech., Acoustics Lab Report Series, Report no. 37, 1995. Available on-line at http://www.acoustics.hut.fi/publications/.

[68] V. Välimäki. Physics-based modeling of musical instruments. In *Proc. Stockholm Music Acoustics Conference*, pages 361–364, Stockholm, Sweden, August 6-9, 2003.

[69] J. Woodhouse. Idealised models of a bowed string. *Acustica*, 79:233–250, 1993.