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

Janne Riionheimo

Parameter Estimation of a Plucked String Synthesis Model via the Genetic Algorithm

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

Espoo, Aug 16, 2004

Supervisor:	Professor Vesa Välimäki
Instructor:	Professor Vesa Välimäki

HELSINKI UNIVERSITY OF TECHNOLOGY

ABSTRACT OF THE MASTER'S THESIS

Author:	Janne Riionheimo
Työn nimi:	Parameter Estimation of a Plucked String Synthesis Model via the Genetic Algorithm
Date:	Aug. 16, 2004 **Number of pages:** 77
Department:	Electrical and Communications Engineering
Professorship:	S-89
Supervisor:	Prof. Vesa Välimäki
Instructors:	Prof. Vesa Välimäki

The aim of this thesis was to develope a method for adjusting the parameters of an existing plucked string synthesis model in such way that the final sound output is perpectually similar to the sound of a real instrument. The existing model has been intensively used for sound synthesis of various string instruments but the fine tuning of the parameters has been carried out with a semiautomatic method that requires some hand adjustment with human listening. By means of the method described in this thesis the parameters of the string model can be now adjusted automatically.

In this thesis previously recorded tones are used as a target with which the synthesized tones are compared. All synthesized tones are then ranked according to their perceptual error value. The perceptual error value is calculated with a method that simulates human hearing and takes its limitations such as frequency dependence and frequency masking into account. The aim is to find a synthesized tone with minimal perceptual error value. In this thesis a genetic algorithm is used to find the minimum.

First, this thesis introduces the plucked string synthesis model and its parameters. Then, the principle of a genetic algorithm and different operators are explained. Thereafter, the calculation of perceptual error value is described. Discretization of the parameters and the implementation of the parameter estimation algorithm are then explained and finally the experimentation and results are shown.

The method described in this thesis enables high quality of synthesis with the plucked string synthesis model and also illustrates the behaviour of the model and clarify how the parameters affect to the final sound output.

Keywords: musical acoustics, sound synthesis, physical modeling synthesis, plucked string synthesis, parameter estimation, genetic algorithm

TEKNILLINEN KORKEAKOULU

DIPLOMITYÖN TIIVISTELMÄ

Tekijä:	Janne Riionheimo
Työn nimi:	Parameter Estimation of a Plucked String Synthesis Model via the Genetic Algorithm
Päivämäärä:	16.8.2004 **Sivuja:** 77
Osasto:	Sähkö- ja tietoliikennetekniikka
Professuuri:	S-89
Työn valvoja:	Prof. Vesa Välimäki
Työn ohjaajat:	Prof. Vesa Välimäki

Tämän diplomityön tavoitteena on ollut kehittää menelmä, jonka avulla on mahdollista säätää soivan kielen synteesimallin parametrit siten, että lopputuloksena on aidon kielisoittimen kuuloinen ääni. Kyseistä kielimallia on käytetty intensiivisesti eri kielisoittimien äänisynteesiin, mutta puoliautomaattisella menetelmällä tehty parametrien hienosäätö on vaatinut käyttäjältä harjaantuneisuutta ja tarkkaa kuuntelukykyä. Tässä diplomityössä esitellyn menetelmän avulla kielimallin parameterit voidaan säätää automaattisesti.

Tämän diplomityön parametrien estimointimenetelmässä käytetään aikaisemmin äänitettyjä kielisoittimien ääniä tavoiteääninä, joihin synteesimallin tuottamia ääniä verrataan. Syntetisoidut äänet järjestetään niiden perkeptuaalisen virheen avulla. Perkeptuaalinen virhe lasketaan menetelmällä, joka simuloi ihmisen kuuloaistia ja huomioi kuuloaistin rajoitteet kuten taajuusriippuvuuden ja peittoilmiön. Tavoitteena on löytää syntetisoitu ääni jolla on mahdollisimman pieni perkeptuaalinen virhearvo. Tämän minimin löytämiseksi käytetään geneettistä algoritmia.

Tässä diplomityössä käsitellään ensin synteesiin käytettävän kielimallin toiminta ja estimoitavat parametrit. Tämän jälkeen kuvataan estimointiin tarvittavan geneettinen algoritmin toiminta ja laskentaan vaikuttavat tekijät. Seuraavaksi esitellään perkeptuaaliseen virheenlaskentaan käytettävä funktio ja sen kuuloaistia simuloivat ominaisuudet. Tämän jälkeen käsitellään parametrien diskretointi sekä esitellään parameterien estimoinnin lopullinen toteutus ja viimeisenä käydään läpi koejärjestelyt sekä tulokset.

Diplomityössä käsitelty menetelmä mahdollistaa parempilaatuisen äänisynteesin kyseisellä kielimallilla ja antaa myös tietoa kielimallin toiminnasta sekä parametrien suhteellisesta vaikutuksesta lopulliseen ääneen.

Avainsanat: musiikkiakustiikka, äänisynteesi, fysikaalinen mallinnus, kielimalli, parametrien estimointi, geneettinen algoritmi

Acknowledgements

This thesis has been prepared in Helsinki University of Technology, at the Laboratory of Acoustics and Audio Signal Processing, during years 2002-2003.

First I would like to deeply thank Professor Vesa Välimäki for instructing and supervising this thesis and for the valuable guidance during my working at the acoustics laboratory. I would also like to thank Dr. Cumhur Erkut, who read through this thesis and gave me helpful notes. My thankfulness also goes to Professor Matti Karjalainen for the pictures and ideas.

I wish to thank everybody in the acoustics lab for the good time and cosy working enviroment.

Finally, I would like to express my gratitude to my parents who have inspired and supported me in various ways during my studies.

Helsinki, August 14, 2004

Janne Riionheimo

Contents

List of Symbols

a_h	Frequency dependent gain of the horizontal string model
a_v	Frequency dependent gain of the vertical string model
b	Degree of non-uniformity
c	Arithmetical crossover parameter
$B(\nu)$	Spreading function
$B_h(\nu)$	Upper boundary of the critical band ν
$B_l(\nu)$	Lower boundary of the critical band ν
B_L	Lower boundary of the parameter range
B_U	Upper boundary of the parameter range
d_f	Difference of the fundamental frequencies
E	Error
E_p	Perceptual error
f	Frequency in Hertz
f_0	Fundamental frequency of the basic string model
$\hat{f}_0$	Fundamental frequency estimate
f_0'	Mean fundamental frequency
$f_{0,h}$	Fundamental frequency of the horizontal string model
$f_{0,v}$	Fundamental frequency of the vertical string model
f_s	Sampling rate
F	Fitness
$F(z)$	Fractional delay filter
F_p	Perceptual fitness
g_c	Coupling gain of the two polarizations
g_h	Loop gain of the horizontal string model
g_v	Loop gain of the vertical string model
H	Hop size or time advance in samples
$H(z)$	Lowpass filter
L	Length of the STFT sequence
L_d	Loop delay
L_f	Fractional part of the loop delay
L_I	Integer part of the loop delay
L_w	Length of the STFT window function
m_o	Output mixing coefficient
m_p	Input mixing coefficient
$M(z)$	Dual-polarization plucked string synthesis model
N	Length of the discrete Fourier transform (DFT)
N_b	Number of bits
N_c	Number of critical bands
N_g	Number of generations
N_k	Number of individuals in tournament selection
$N_p(\nu)$	Number of points in each Bark band ν

$o(n)$	Output sound of the model
p	Normalizing gain
p_c	Probability of crossover
p_m	Probability of mutation
$P(k)$	Power spectrum
P_i	Probability of selecting the ith individual
q	Probability of selecting the best individual
$Q(\nu)$	Normalized energy
r	Rank of the individual
r_p	Range of variation of f_0'
$R(\nu)$	Raw masking threshold
$S(z)$	Basic string model
$S_h(z)$	Horizontal string model
$S_v(z)$	Vertical string model
S_p	Population size
$S_P(\nu)$	Spread energy per critical band
S_t	Truncation parameter
$t(n)$	Target sound
$U(\nu)$	Masking energy offset
V	Spectral flatness measure
$w(n)$	Window function
w	Number of attempts in heuristic crossover
$W(k)$	Final threshold of masking
$W_s(k)$	Inverted equal loudness curve at the SPL of 60 dB
$W_{f,m}$	Weighting matrix
$W_{t,m}$	Weighting matrix
$\vec{x}$	Chromosome represented as a vector array
$y(n)$	Example signal
$Y_m(k)$	STFT of signal $y(n)$
z^{L_I}	Delay line
$Z(\nu)$	Energy per critical band
α	Tonality factor
δ	The difference between the level of the masker and the masking threshold
ν	Frequency in Bark units
τ	Time constant of the overall decay

List of Abbreviation

DFT	Discrete Fourier transform
FFT	Fast Fourier transform
GA	Genetic algorithm
GAOT	The Genetic Algorithm Optimization Toolbox for Matlab 5
MSE	Mean squared error
SMR	Signal-to-mask ratio
SPL	Sound pressure level
STFT	Short-time Fourier transform

Chapter 1

Introduction

The development of a parameter estimation procedure for an existing plucked string synthesis model is described in this thesis. By means of the procedure the control parameters of the synthesis model can be adjusted such a way that the sound output is perceptually similar to the sound of a real instrument. The procedure in a nutshell is as follows. The synthesis model is first used to produce a set of tones that are compared to a previously recorded real instrument tone that functions as a target of the procedure. The perceptual dissimilarity between the tones, presented as an error value, is determined and the worst individuals are dumped. Parameter values are then varied, another set of synthesized tones is produced, and error values are calculated. This is proceeded until the sound output of the model is indistinguishable from the recorded tone.

1.1 Background

Physical modeling of musical instruments has become one of the most important research topics in the field of sound synthesis these days. Physical modeling synthesizers have been successful, which is no wonder since model-based sound synthesis is a powerful tool for creating natural sounding tones by simulating the sound production mechanisms and physical behavior of real musical instruments. A strength of physically modelled instruments is that the control parameters are easy to understand because they have a real counterpart. Besides real acoustical instruments, a hot trend in the music industry is the modeling of analog techniques used previously for sound synthesis. Imperfections of these retro instruments have been noticed to bring liveliness and human characteristics to synthesized sound.

Roads sees the goals of physics-based sound synthesis as scientific one and artistic one (Roads, 1996). The goal of the scientific approach is to gain understanding of the physical behavior of real instruments. These sound production mechanisms are often too complex to simulate in every detail, and therefore simplified models are designed for synthesis. Generated models, which are perceptually indistinguishable from real instruments, are then used for artistic purposes. Also non-existing virtual instruments can be constructed with a computer and utilized in composing.

Parameter estimation for such systems is an important and difficult challenge. Usually the natural parameter settings are in great demand at the initial state of the synthesis. When using these parameters with a model, we are able to produce real-sounding instrument tones. Various methods for adjusting the parameters to produce desired sounds have been proposed in the literature. Calibration of a plucked string guitar synthesis model and extraction of expressive parameters is proposed in (Välimäki et al., 1996; Tolonen and Välimäki, 1997; Erkut et al., 2000). Parameter estimation for dual-polarization plucked string models using a sub-band Hankel singular value decomposition algorithm is described in (Nackaerts et al., 2001). Suitable synthesis parameters are determined by employing the learning ability of neural networks (Liang and Su, 2000) of a system where the nonlinear component is modelled by radial basis function networks (Drioli and Rocchesso, 1998). My interest in this thesis is the parameter estimation of the model proposed by Karjalainen et al. (Karjalainen et al., 1998). An automated parameter estimation method for the model, known as *Calibrator*, has been proposed in (Välimäki et al., 1996; Tolonen and Välimäki, 1997), and then improved in (Erkut et al., 2000). By means of *Calibrator* the parameters of the model have been earlier estimated automatically, but the fine-tuning have required some hand adjustment.

In this thesis, recorded tones are used as a target with which the synthesized tones are compared. All possible synthesized sounds are then ranked according to their similarity with the recorded tone. An accurate way to measure sound quality from the viewpoint of auditory perception would be to carry out listening tests with trained participants and rank the candidate solutions according to the data obtained from the tests (Mattila and Zacharov, 2001). This method is extremely time consuming and therefore we are forced to use analytical methods to calculate the quality of the solutions. An error function that simulates the human hearing and calculates the perceptual error between the tones is developed in this thesis. Frequency masking behavior, frequency dependence, and other limitations of human hearing are taken into account. From the optimization point of view the task is to find the global minimum of the error function. The variables of the function, i.e. the parameters of the synthesis model, span the parameter space where each point corresponds to a set of parameters and thus to a synthesized sound. When various parameters are used to control the model the parameter space expands and the optimization of the error function becomes a difficult task, where specialized methods are needed. In this thesis a genetic algorithm is used

to optimize the perceptual error function.

1.1.1 Thesis Outline

This thesis is sectioned in three parts, the first of which (Chapters 2-3) concentrates on theory of the plucked string synthesis model and genetic algorithms. The control parameters to be estimated as well as possible methods for estimation are described in Chapter 2. The principle of a genetic algorithm and different operators are explained in Chapter 3.

The second part of the thesis (Chapters 4-5) is about the implementation of the procedure. Chapter 4 concentrates on the calculation of the perceptual error. In Chapter 5 the parameter space is discretized in a perceptually reasonable manner and the final implementation of the parameter estimation procedure is explained. The third part consists of Chapter 6 where the parameter estimation procedure is tested and results are analyzed. Conclusions are finally drawn in Chapter 7.

Chapter 2

Physical Modeling Synthesis of a Plucked String

A workable method for physical modeling synthesis is based on digital waveguide theory proposed by Smith (Smith, 1992). In the case of the plucked string instruments the method can be extended to model also the plucking style and instrument body (Smith, 1993; Karjalainen et al., 1993). A synthesis model of this kind can be applied to synthesize various plucked string instruments by changing the control parameters and using different body and plucking models (Välimäki et al., 1996; Laurson et al., 2001). A characteristic feature in string instrument tones is the double decay and beating effect (Weinreich, 1977), which can be implemented by using two slightly mistuned string models in parallel to simulate the two polarizations of the transversal vibratory motion of a real string (Karjalainen et al., 1998). A synthesis model of this kind is introduced in the beginning of this chapter, after which the possible parameter estimation procedures for the model are surveyed.

2.1 Plucked String Synthesis Model

The model proposed by Karjalainen et al. (Karjalainen et al., 1998) is used for plucked string synthesis in this thesis. The block diagram of the model is presented in Figure 2.1. It is based on digital waveguide synthesis theory (Smith, 1992) that is extended in accordance with the commuted waveguide synthesis approach (Smith, 1993; Karjalainen et al., 1993) to include also the body modes of the instrument to the string synthesis model.

Different plucking styles and body responses are stored as wavetables in the memory

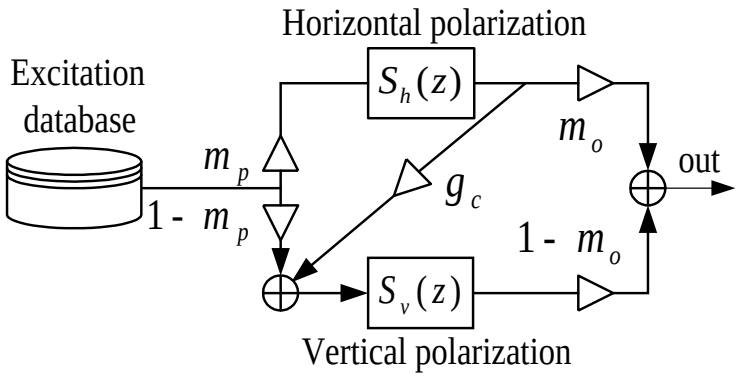


Figure 2.1: The plucked string synthesis model.

and used to excite the two string models $S_h(z)$ and $S_v(z)$ that simulate the effect of the two polarizations of the transversal vibratory motion. A single string model $S(z)$ in Figure 2.2 consists of a lowpass filter $H(z)$ that controls the decay rate of the harmonics, a delay line z^{-L_I}, and a fractional delay filter $F(z)$. The delay time around the loop for a given fundamental frequency f_0 is

$$L_d = \frac{f_s}{f_0}, \tag{2.1}$$

where f_s is the sampling rate (in Hz). The loop delay L_d is implemented by the delay line z^{-L_I} and the fractional delay filter $F(z)$. The delay line is used to control the integer part L_I of the string length while the coefficients of the filter $F(z)$ are adjusted to produce the fractional part L_f (Jaffe and Smith, 1983; Laakso et al., 1996). $F(z)$ is implemented as a first-order all-pass filter. Two string models are typically slightly mistuned to produce a natural sounding beating effect.

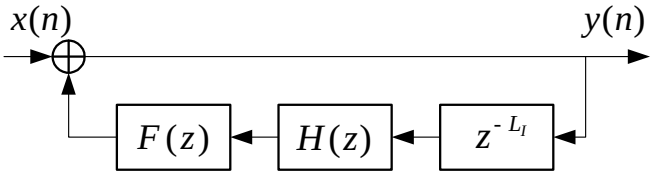


Figure 2.2: The basic string model.

A one-pole filter with transfer function

$$H(z) = g\frac{1+a}{1+az^{-1}} \tag{2.2}$$

is used as a loop filter in the model. Parameter $0 < g < 1$ in Eq. 2.2 determines the overall decay rate of the sound while parameter $-1 < a < 0$ controls the frequency-dependent decay. The excitation signal is scaled by the mixing coefficients m_p and $(1 - m_p)$ before sending it to two string models. Coefficient g_c enables coupling between the two polarizations. Mixing coefficient m_o defines the proportion of the two polarization in the output sound. All parameters m_p, g_c and m_o are chosen to have values between 0 and 1. The transfer function of the entire model is written as

$$\begin{aligned} M(z) &= m_p m_o S_h(z) + (1-m_p)(1-m_o)S_v(z) + \\ &\quad + m_p(1-m_o)g_c S_h(z)S_v(z), \end{aligned} \tag{2.3}$$

where the string models $S_h(z)$ and $S_v(z)$ for the two polarizations can be written as an individual string model

$$S(z) = \frac{1}{1 - z^{-L_I}F(z)H(z)}. \tag{2.4}$$

The model in Figure 2.1 can be rearranged and presented as in Figure 2.3.

If Equations 2.3 and 2.4 without a fractional delay filter $F(z)$ are combined we get an expanded transfer function for the model as follows

$$\begin{aligned} M(z) &= \frac{M_s + M_s a_s z^{-1} + M_s a_m z^{-2} - M_2 G_h z^{-L_{Ih}} - M_1 G_v z^{L_{Iv}} -}{1 + a_s z^{-1} + a_m z^{-2} - G_v z^{-L_{Iv}} - G_h z^{-L_{Ih}} - G_v a_h z^{(-L_{Iv}-1)} -} \cdots \\ &\quad \cdots \frac{-M_2 G_h a_v z^{(-L_{Ih}-1)} - M_1 G_v a_h z^{(-L_{Iv}-1)}}{-G_h a_v z^{(-L_{Ih}-1)} + G_h G_v z^{(-L_{Ih}-L_{Iv})}}, \end{aligned} \tag{2.5}$$

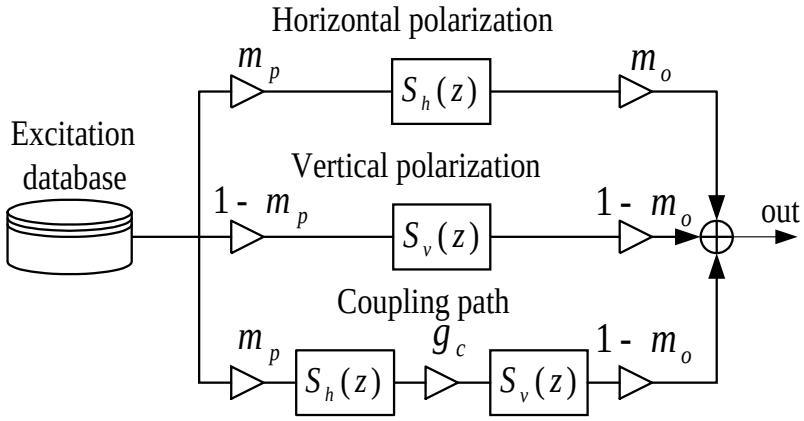


Figure 2.3: The rearranged plucked string synthesis model.

where

$$\begin{aligned} M_1 &= m_o m_p \\ M_2 &= (1-m_o)(1-m_p) \\ M_3 &= g_c m_p (1-m_o) \\ M_s &= M_1 + M_2 + M_3 \\ M_m &= M_1 M_2 M_3 \\ a_s &= a_h + a_v \\ a_m &= a_h a_v \\ G_h &= g_h(1+a_h) \\ G_v &= g_v(1+a_v) \end{aligned}$$

Equation 2.5 is useful when analyzing the effect of coupling and mixing parameters as well as orthogonality of the model. Although the model appears to be asymmetric for two polarizations due to unidirectional coupling path we can define m_p and m_o such a way that the parameters of individual string model are exchangeable between polarizations. It can be seen in Figure 2.3 that if

$$m_p = 1 - m_o$$

the transfer functions $S(z)_h$ and $S(z)_v$ of individual polarizations can be swapped without affecting to the transfer function $M(z)$ of the entire model. This implies that similar tones are produced with parameter sets of two individual string model regardless of which way the sets are ordered.

Synthesis model of this kind has been intensively used for sound synthesis of various plucked string instruments (Laurson et al., 2001; Erkut et al., 2001; Erkut and Välimäki, 2000). Different methods for estimating the parameters have been used, but in consequence of interaction between the parameters, systematic methods are at least troublesome but probably impossible. The nine parameters that are used to control the synthesis model are listed in Table 2.1.

parameter	control
$f_{0,h}$	fundamental frequency of the horizontal string model
$f_{0,v}$	fundamental frequency of the vertical string model
g_h	loop gain of the horizontal string model
a_h	frequency dependent gain of the horizontal string model
g_v	loop gain of the vertical string model
a_v	frequency dependent gain of the vertical string model
m_p	input mixing coefficient
m_o	output mixing coefficient
g_c	coupling gain of the two polarizations

Table 2.1: Control parameters of the synthesis model.

2.2 Estimation of the Model Parameters

Determination of the proper parameter values for sound synthesis systems is an important problem and also depends on the purpose of the synthesis. When the goal is to imitate the sounds of real instruments the aim of the estimation is unambiguous: We wish to find a parameter set which gives the sound output that is sufficiently similar with the natural one in terms of human perception. These parameters are also feasible for virtual instruments at the initial stage after which the limits of real instruments can be exceeded by adjusting the parameters in more creative ways.

Parameters of a synthesis model correspond normally to the physical characteristics of an instrument (Karjalainen et al., 1998). The estimation procedure can then be seen as sound analysis where the parameters are extracted from the sound or from the measurements of physical behavior of an instrument (Roads, 1996). Usually, the model parameters have to be fine-tuned by laborious trial and error experiments, in collaboration with accomplished players (Roads, 1996). Parameters for the synthesis model in Figure 2.1 have been earlier estimated this way and recently in a semi-automatic fashion, where some parameter values can be obtained with an estimation algorithm while others must be guessed. Another approach is to consider the parameter estima-

tion problem as a nonlinear optimization process and take advantage of the general searching methods. All possible parameter sets can then be ranked according to their similarity with the desired sound.

2.2.1 Sound Analysis Based Parameter Estimation

Calibrator

A brief overview of the calibration scheme used earlier with the model in (Välimäki et al., 1996; Tolonen and Välimäki, 1997) is given here. The block diagram from (Erkut et al., 2000) is presented in Figure 2.4. The fundamental frequency $\hat{f}_0$ is first estimated using the autocorrelation method. The frequency estimate in samples from Equation 2.1 is used to adjust the delay line length L_I and the coefficients of the fractional delay filter $F(z)$. The amplitude, frequency, and phase trajectories for partials are analyzed using the short-time Fourier transform (STFT), as in (Välimäki et al., 1996). The estimates for loop filter parameters g and a are then analyzed from the envelopes of individual partials.

The excitation signal for the model is extracted from the recorded tone by a method described in (Välimäki and Tolonen, 1998). The amplitude, frequency, and phase trajectories are first used to synthesize the deterministic part of the original signal and the residual is obtained by a time-domain subtraction. This produces a signal which lacks the energy to excite the harmonics when used with the synthesis model. This is avoided by inverse-filtering the deterministic signal and the residual separately. In (Erkut et al., 2000) an optimization technique for the procedure is proposed. Output signal of the model is fed to the routine which automatically fine-tunes the model parameters by analyzing the time-domain envelope of the signal.

Extracting Beating

The difference in the length of the delay lines can be estimated based on the beating of a recorded tone. In (Välimäki et al., 1999) the beating frequency is extracted from the first harmonic of a recorded string instrument tone by fitting a sine wave using the least squares method. According to Välimäki et al. this yields a good estimate of the difference in the fundamental frequencies of the two polarization but the method is probably useful only for instruments with strong beating characteristics, such as the kantele.

Another procedure for extracting beating and two stage decay from the string tones

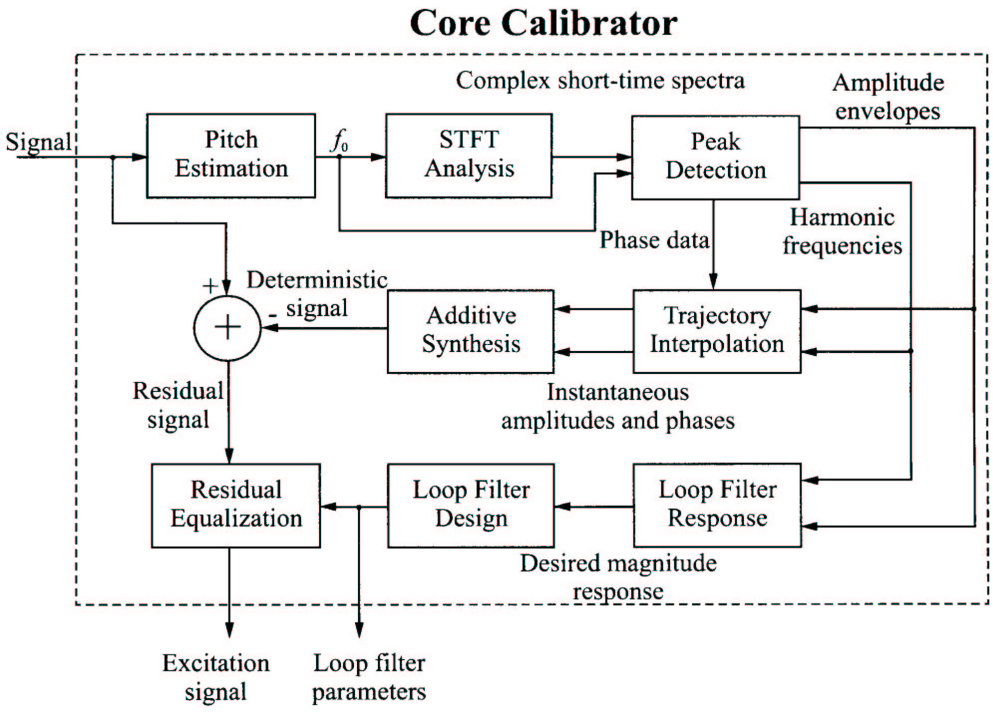


Figure 2.4: Block diagram of the calibrator from (Erkut et al., 2000).

is described by Bank in (Bank, 2000). The general exponential decay is first removed from the amplitude envelope of one particular partial. The resulting signal, describing the deviation from the ideal exponential decay, is then normalized and a model of an exponentially rising or decaying sinusoid is fit on the data. The amplitude and decay time of the normalized deviation signal is computed by a similar method that was used to analyze the partial envelopes. Each partial is analyzed separately and beating and two stage decay is realized with a model consisting of one string model and a resonator bank instead of two parallel string models (Bank et al., 2000).

In practice the mistuning between the two string models has been found by ear (Laurson et al., 2001).

2.2.2 Optimization Based Parameter Estimation

Instead of extracting the parameters from audio measurements an approach is to find the parameter set that produces a tone that is perceptually indistinguishable from the target one. Each parameter set can be assigned with the quality value, which denotes how good the candidate solution is. This performance metric is usually called a fitness function, or inversely, an error function. A parameter set is fed into the fitness function

which calculates the error between the corresponding synthesized tone and the desired sound. The smaller the error the better the parameter set and the higher the fitness value. These functions give a numerical grade to possible solution by means of which we are able to classify candidate parameter sets.

When dealing with discrete parameter values the number of parameter sets is finite and given by the product of the numbers of possible values of each parameter. Using nine control parameters with 100 possible values a total of 10^{18} combinations exist in the space and therefore an exhaustive search is obviously impossible. Therefore, more effective optimizing algorithms have to be used.

Evolutionary algorithms have shown a good performance in optimizing problems relating to the parameter estimation of synthesis models. Vuori and Välimäki (Vuori and Välimäki, 1993) tried a simulated evolution algorithm for the flute model and Horner et al. proposed an automated system for parameter estimation of FM synthesizer using a genetic algorithm (Horner et al., 1993). Genetic algorithms have been used for designing sound synthesis algorithms automatically in (Garcia, 1998; Johnson, 1999). In this thesis a genetic algorithm is used to optimize the perceptual error function.

Chapter 3

Genetic Algorithm

Genetic algorithms (GA) mimic the evolution of the nature and take advantage of the principle of survival of the fittest (Mitchell, 1998). These algorithms operate on a population of potential solutions improving characteristics of the individuals from generation to generation. Each individual, called a chromosome, is made up of an array of genes that contain in our case the actual parameters to be estimated. In every generation, a new set of chromosomes is created by crossing and mutating the fittest individuals from the previous generation. Desired characteristics are passed into the next generation while imperfect individuals become extinct.

Genetic algorithms have been developed by John Holland and his students and colleagues at the University of Michigan in the 1960s and 1970s. Since then the genetic algorithms have been intensively used in various disciplines for solving numerous optimizing problems. Genetic algorithms are theoretically and empirically proven to provide robust search especially in complex spaces (Goldberg, 1989).

In this chapter we discuss first the basic features of genetic algorithms. To illustrate the behavior of an algorithm we use the original algorithm design for a simple optimization problem: to find a maximum of a single variable function with known maximum points. Possible improvements for the simple GA are surveyed and the operators, which are later in this book applied to our algorithm, are explained in detail.

3.1 Operation of a Genetic Algorithm

Genetic algorithm operates on the population of S_p individuals. Each individual represents a potential solution to the problem. All the individuals in a generation are

evolved in parallel which is a clear advantage if various local minima exist in the search space. In the original algorithm design the chromosomes were represented with binary digits (Holland, 1975). Michalewitz introduced the floating point representation of GA and developed operators for the algorithm (Michalewicz, 1992). He also described the characteristics of the operators.

A simple algorithm is implemented as follows:

1. Initialization: Create a random population of S_p individuals (chromosomes).
2. Fitness calculation: Calculate the fitness for each individual in the initial population.

   Repeat the following steps until termination

3. Selection: Select individuals from the current population to the mating pool to produce a new generation.
4. Altering: Alter the new population by crossing and mutating the individuals.
5. Evaluate: Calculate the fitness for each individual.
6. Replace the current population with the new one.

The algorithm is normally terminated when a specified number of generations is produced or when the sum of the deviations among the individuals becomes smaller than some specified threshold. Several schemes are feasible for selection, crossover and mutation processes depending on which chromosome representation is used.

3.2 Finding a maximum of a Simple Function

In the following example we use a genetic algorithm to find a maximum of a simple function $f(x)$ with one variable x. The aim is to illustrate the behavior of a GA at grass roots. All the steps are implemented in a simplified manner but realistically.

The function is defined as follows

$$f(x) = -x^2 + 24\sin(x) + 120, \tag{3.1}$$

where $x = [-10, 10]$. The function graph is drawn in Figure 3.1.

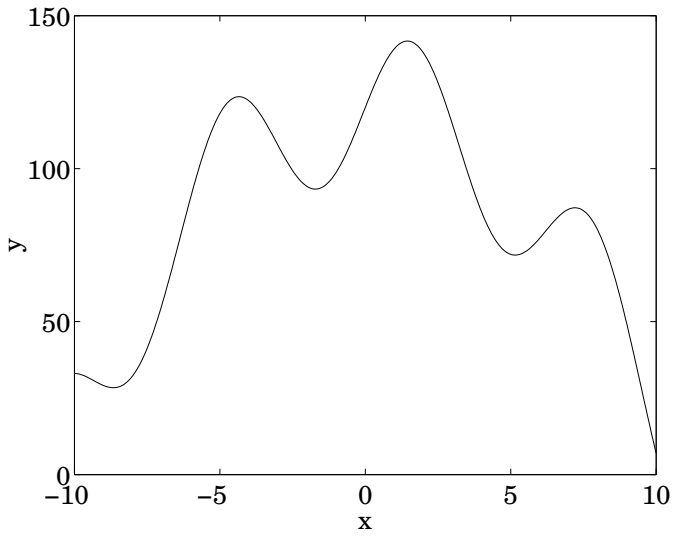


Figure 3.1: Graph of the function $f(x) = -x^2 + 24\sin(x) + 120$.

The maxima of the function in the specified range are analyzed by determining the zeros of the first derivative $f(x)'$ for the function $f(x)$

$$f(x)' = -2x + 24\cos(x) = 0. \tag{3.2}$$

The zeros are determined by iteration and the approximate maxima are found at $x = -4.3421, x = 1.4497$, and $x = 7.2095$. Maximum function value is found in $x = 1.4497$ where $f(1.4497) = 141.7226$.

3.2.1 Binary Representation

In binary representation the precision of the solution depends on the number of bits in a binary number. Genetic algorithm manipulates the strings of binary digits, where each bit is considered as a gene of a chromosome.

Discretizing the range [-10,10] with the precision of 0.0001 implies that at least 200000 uniformly distributed values should be used for the range. Representing this with binary numbers means that 18 bits are required. The mapping from 18 bit binary number into the real number from the range [-10,10] is carried out by first converting

a binary string to a real number, for example

$$x_b = (011001011001001011)_{base2} = 104011_{base10},$$

which is an index number signifying 104011th value in a discrete grid. A corresponding real number from the range [-10,10] is then calculated as follows

$$x = -B_L + x_b \frac{B_U - B_L}{2^{N_b} - 1}, \tag{3.3}$$

where the B_U is the upper and B_L is the lower boundary of the range (10 and -10 in our example) and N_b is the number of bits (18 in our example). The example binary digit x_b represents therefore the real number -2.0646.

3.2.2 Initialization

Random population of $S_p = 10$ individuals shown in Table 3.1 is first created for optimizing the function $f(x)$. The binary numbers are converted from base two to base ten and the corresponding real values are calculated with Eq. 3.3. The function value represents the fitness of each chromosome. We use a small population size to get a sparse distribution over the variable x at the initial state. In order to illustrate the behavior of a genetic algorithm the best possible initial population is not required.

No.	chromosome	corresponding real number	function value
1	101000011001011001	2.6240	124.9889
2	110100010110110010	6.3613	81.4067
3	110011110100101111	6.1951	79.5089
4	010010100011011000	-4.2022	123.2853
5	001100001000100011	-6.2083	83.2543
6	001110100011101110	-5.4506	108.0438
7	000011101000110011	-8.8633	28.6634
8	000101111000011011	-8.1620	30.5113
9	100000011011101001	0.1350	123.2120
10	011101100000010001	-0.7799	102.5144

Table 3.1: Initial population of ten chromosomes, corresponding real numbers, and function values

3.2.3 Selection

Selection procedure plays an important role in genetic algorithm. The idea is to give preference to better individuals and allow them to pass to the mating pool where the individuals are altered before they form up a new generation. Chromosomes are selected according to their fitness values such that the chromosomes with higher value have an increased probability of contributing one or more offspring in the next generation. Several schemes are available for the selection process: roulette wheel, elitist, tournament, scaling techniques, linear and nonlinear ranking methods, and truncation selection.

We use here the roulette wheel, which was the first selection method developed by Holland. Each chromosome has a probability of being selected that is determined by the chromosome's fitness as a percentage of the total population fitness. Each chromosome has a roulette wheel slot that is sized according to the chromosomes proportion. The roulette wheel with slots according to our initial population in Table 3.1 is shown in Figure 3.2.

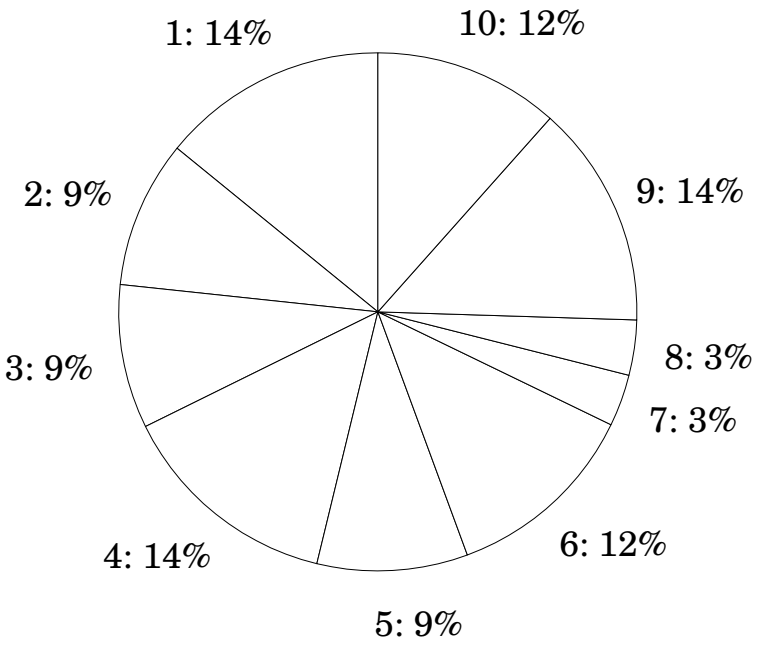


Figure 3.2: Weighed roulette wheel with slots, size of which depend on the fitness value of corresponding chromosome. Slots are numbered according to chromosomes in Table 3.1

Roulette wheel is spun S_p times to select individuals to the mating pool. Spinning is simulated by generating S_p random numbers pointing to the wheel. Obviously, some chromosomes would be selected more than once. The worst chromosomes die off while the better ones get more copies and therefore the average fitness will increase

throughout the process.

The chromosomes in Table 3.2 are selected from our initial population by the roulette wheel scheme. As we can see the poorest chromosomes number 3,7 and 8 with function values 79.5089, 28.6634, and 30.5113 in the initial population in Table 3.1 are replaced by the copies of the better chromosomes and the average fitness value has increased.

No.	chromosome	corresponding real number	function value
1	101000011001011001	2.6240	124.9889
2	101000011001011001	2.6240	124.9889
3	110100010110110010	6.3613	81.4067
4	010010100011011000	-4.2022	123.2853
5	001100001000100011	-6.2083	83.2543
6	001110100011101110	-5.4506	108.0438
7	001110100011101110	-5.4506	108.0438
8	001110100011101110	-5.4506	108.0438
9	100000011011101001	0.1350	123.2120
10	011101100000010001	-0.7799	102.5144

Table 3.2: Mating pool. Chromosomes are selected with roulette wheel from the initial population in Table 3.1

Chromosomes in the mating pool are then altered by crossover and mutation operations.

3.2.4 Altering

Two classical genetic operators are available for the altering: crossover and mutation. Crossover operator picks two parents and produces offspring by splitting the parents in a random point and swapping the parts. Crossing the two chromosomes $\vec{x_{b,3}}$ and $\vec{x_{b,5}}$ from the mating pool

$$\vec{x_{b,3}} = (1101|00010110110010)$$
$$\vec{x_{b,5}} = (0011|00001000100011)$$

with fitness values

$$f(\vec{x_{b,3}}) = f(6.3613) = 81.4067$$
$$f(\vec{x_{b,5}}) = f(-6.2083) = 83.2543$$

results in the two offspring vectors

$$\vec{x_{b,3}}' = (1101|00001000100011)$$
$$\vec{x_{b,5}}' = (0011|00010110110010)$$

with fitness values

$$f(\vec{x_{b,3}}') = f(6.2918) = 80.6200$$
$$f(\vec{x_{b,5}}') = f(-6.1387) = 85.7720.$$

Each chromosome in the mating pool has a chance of being a parent for crossing, which is controlled by the probability of crossover p_c. Only a even number of parents can be selected.

Mutation flips each bit in every chromosomes with some specified probability p_m. Mutating two genes (fifth and tenth) in a chromosome $\vec{x_{b,10}}$

$$\overrightarrow{x_{b,10}} = (0111\mathbf{0}11000\mathbf{0}0010001)$$

with the fitness $f(\vec{x_{b,10}}) = f(-0.7799) = 102.5144$ produces a chromosome $\vec{x_{b,10}}'$

$$\overrightarrow{x_{b,10}}' = (0111\mathbf{1}11001\mathbf{1}0010001)$$

with the fitness $f(\vec{x_{b,10}})' = f(-0.1354) = 116.7424$. In this particular case the mutation and crossover procedures improve the chromosome $\vec{x_{b,10}}$ and $\vec{x_{b,5}}$ while the fitness value for the offspring vector $\vec{x_{b,3}}'$ is lower than in its parent $\vec{x_{b,3}}$.

3.2.5 Test Run

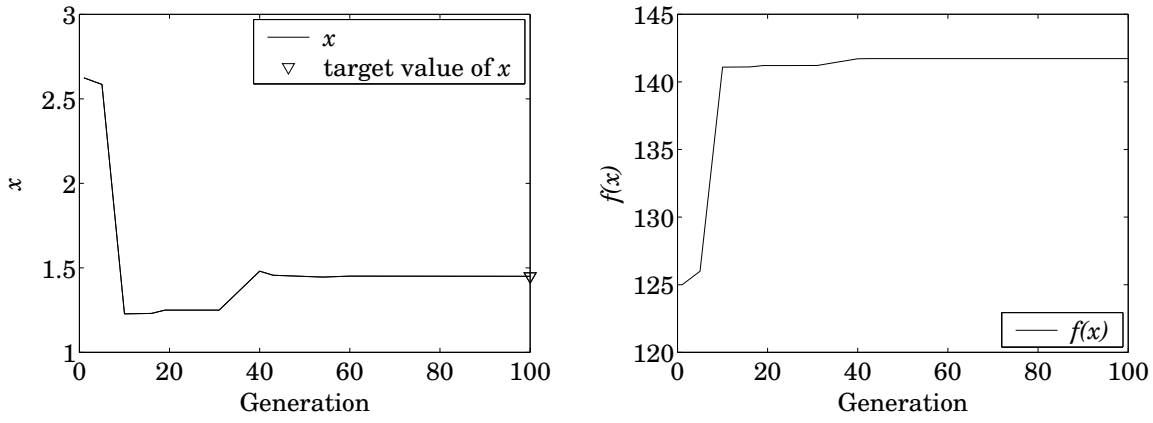
We have run GA with the following parameters:

Population size $S_p = 10$, number of generations $= 100$, probability of crossover $p_c = 0.6$, probability of mutation $p_m = 0.05$.

The results are shown in Table 3.3 where the generations with improvement in the function value are presented. The exact global maximum is found at $x = 1.4497$. The corresponding binary digit is $(1001001010011100)_{base2}$. The convergence of x and $f(x)$ is shown in Figure 3.3.

Generation	best real number value	function value
1	2.6240	124.9889
5	2.5850	125.9978
10	1.2278	141.0943
16	1.2302	141.1080
19	1.2497	141.2118
25	1.2499	141.2126
31	1.2499	141.2130
40	1.4802	141.7106
43	1.4558	141.7221
54	1.4454	141.7224
60	1.4507	141.7226
81	1.4497	141.7226
100	1.4497	141.7226

Table 3.3: Generations with improvement in the fitness value.



(a) (b)

Figure 3.3: Convergence of the function variable x and the function value $f(x)$.

3.3 Improvements

One disadvantage of the roulette wheel selection scheme is that it does not guarantee that the fittest chromosome will be selected. Therefore, the global maximum is not necessarily found, especially when a small population size is used. This behavior can be avoided by using other selection methods:

- **Elitist models**: According to Goldberg (1989) the first improved selection scheme was proposed by De Jong in (De Jong, 1975). The best chromosome is enforced to survive to the next generation. De Jong designed five variations of the basic GA.
- **Tournament selection** (Goldberg et al., 1991): Some number N_k of individuals is chosen randomly from the current population and the best individual from this group is selected into the next generation. This process is repeated S_p number of times. The values for N_k ranges from 2 to S_p. The typical value is $N_k = 2$.
- **Ranking** (Grefenstette and Baker, 1989; Goldberg, 1989): Individuals are first sorted according to their fitness and given a rank number r, where $r = 1$ is the best and $r = S_p$ is the worst. Probability of selecting the rth individual to the next generation is then calculated by a linear function (linear ranking), e.g.

$$P(r) = q - (r-1)r, \tag{3.4}$$

where

$$r = \frac{q}{S_p - 1}, \tag{3.5}$$

or a nonlinear function (nonlinear ranking), e.g.

$$P(r) = q(1-q)^{r-1}, \tag{3.6}$$

where $0 < q < 1$ is the user defined parameter which denotes the probability of selecting the best individual. Another possible ranking scheme is proposed in (Joines and Houck, 1994). It is called normalized geometric ranking, where the probability is calculated by the function

$$P(r) = q'(1-q)^{r-1}, \tag{3.7}$$

where

$$q' = \frac{q}{1-(1-q)^{S_p}}. \tag{3.8}$$

Parameter q is also said to control the so called selection pressure. If $q = 0$ all individuals have the same probability of being selected while $q = 1$ means that only copies of the best individual are selected. An example case for the probability of selection for each individual when $S_p = 10$ and $q = 0.05$ is shown in Table 3.4.

rank r	Probability of selection $P(r)$
1	0.1246
2	0.1184
3	0.1125
4	0.1068
5	0.1015
6	0.0964
7	0.0916
8	0.0870
9	0.0827
10	0.0785

Table 3.4: Probability of selection $P(r)$ for a population of ten individuals when normalized geometric ranking with $q = 0.05$ is used.

It is noteworthy that the selection probabilities do not depend on the absolute fitness values but only on the rank.

- **Truncation** (Mühlenbein and Schlierkamp-Voosen, 1993): Individuals are first sorted similarly as in ranking methods. Only the best S_t individuals are selected to mating pool. S_t indicates the proportion of the population to be selected as parents and takes values ranging from 50%-10%. Individuals below the truncation threshold do not produce offspring.

Improvements in crossover and mutation schemes have also been studied by various researchers. Eshelman et al. have experimented with other possible crossover schemes

such as two-point, multi-point, segmented, and shuffle crossover (Eshelman et al., 1989). Two-point and multi-point crossover select multiple crossing points and swap chromosome parts between them. Segmented crossover allows the number of crossover points to vary by replacing the fixed number of crossover points with a switch rate that specifies a probability of any point in a string to be a crossover point. A possible improvement is to change over from binary to the floating point representation.

3.4 Floating Point Representation

In floating point representation each chromosome is coded as a vector of floating point numbers. Each gene contains one number which is forced to be within the desired range. Michalewicz showed that the floating point representation results in faster, more consistent, higher precision, and more intuitive solution of the algorithm (Michalewicz, 1992) especially with large domains.

One possible chromosome with nine components (genes) containing parameters of the plucked string synthesis model is shown in Table 3.5.

$f_{0,h}$	$f_{0,v}$	g_h	a_h	g_v	a_v	m_p	m_o	g_c
330	331	0.9892	-0.21	0.9912	-0.198	0.5	0.5	0.1

Table 3.5: A chromosome with nine genes containing parameters of the plucked string synthesis model.

Crossover and mutation schemes applied to chromosomes with floating point numbers differ from simple ones described above.

3.4.1 Crossover

- **Simple crossover**: Similar than in binary representation. A chromosome is split in a random point and the parts are swapped.
- **Arithmetical crossover**: Produces two offspring $\vec{x}_{o,1}$ and $\vec{x}_{o,2}$ that are linear combinations of the two parents $\vec{x}_{p,1}$ and $\vec{x}_{p,2}$ as follows

$$\vec{x}_{o,1} = c\vec{x}_{p,1} + (1-c)\vec{x}_{p,2} \quad (3.9)$$

$$\vec{x}_{o,2} = (1-c)\vec{x}_{p,1} + c\vec{x}_{p,2} \quad (3.10)$$

where $0 \leq c \leq 1$. The parameter c can be either a constant (uniform arithmetical crossover), or a variable whose value depends on the age of population (non-uniform arithmetical crossover). If $c = 0.5$ the scheme is called a guaranteed average crossover.

- **Heuristic crossover**: Produces a single offspring $\vec{x}_o$ which is a linear extrapolation of the two parents $\vec{x}_{p,1}$ and $\vec{x}_{p,2}$ as follows

$$\vec{x}_o = h(\vec{x}_{p,2} - \vec{x}_{p,1}) + \vec{x}_{p,2}, \tag{3.11}$$

where $0 \leq h \leq 1$ is a random number and the parent $\vec{x}_{p,2}$ is not worse than $\vec{x}_{p,1}$. Non feasible solutions are possible and if no solution after w attempts is found, the operator gives no offspring. Heuristic crossover contributes to the precision of the final solution.

3.4.2 Mutation

- **Uniform mutation**: Sets a randomly selected component (gene) in a chromosome to an uniform random number between the boundaries. In the early phase the operator allows the candidate solutions to move freely around the parameter space. In the later phases the parameter contributes the solution not to be stuck in a local optimum.

- **Non-uniform mutation**: Selects randomly one variable i in a chromosome, and sets it equal to an non-uniform random number as follows:

$$\vec{x_i}' = \begin{cases} \vec{x_i} + (B_U - \vec{x_i}) f(G) & \text{if } h_1 < 0.5, \\ \vec{x_i} - (\vec{x_i} - B_L) f(G) & \text{if } h_1 \geq 0.5, \end{cases}$$

where $0 \leq h_1 \leq 1$ is a random number and B_U and B_L are the upper and lower boundaries of the variable. h_1 chooses the direction of mutation. If $h_1 < 0.5$ the variable is mutated towards the lower bound while $h_1 \geq 0.5$ mutates towards the upper bound. Function $f(G)$ weights the difference between $\vec{x_i}$ and the bound non-uniformly. Function can be defined according to the application for example as follows

$$f(G) \;=\; 1 - h_2^{\left(1 - \frac{G}{N_g}\right)^b},$$

h_2 = a random number between [0,1],
G = current generation,
N_g = number of generations in total,
b = degree of non-uniformity.

The probability for the function $f(G)$ being close to zero increases as G approaches N_g as can be seen in Figure 3.4. This means that the scheme operates uniformly over the space at the early stage of the algorithm. When the current generation approaches the maximum number of generations the operator searches very locally and therefore contributes in fine tuning. The degree of non-uniformity is controlled with the parameter b. This affects the steepness of the curves in Figure 3.4.

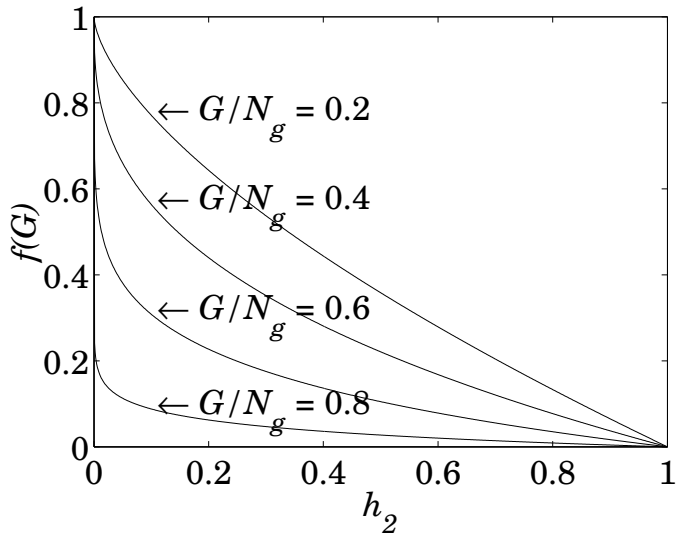


Figure 3.4: $f(G)$ of four selected moments while $b = 2$.

- **Multi-non-uniform mutation**: Changes all genes in the chromosome non-uniformly.
- **Boundary mutation**: Sets a parameter to one of its boundaries. It is useful if the optimal solution is supposed to lie near the boundaries of the parameter space.

Although instructions for particular parameter and operator setting of genetic algorithms can be found in the literature it seems unlikely that any general principles can be formulated but the settings are always application dependant (Mitchell, 1998). The comparison of different selection schemes is reported in (Blickle and Thiele, 1995) and

a study on influence of the control parameters on the GA is presented in (Schaffer et al., 1989).

Chapter 4

Fitness Calculation

Genetic algorithms are powerful tools for searching large and complicated spaces. The application areas vary from learning algorithms for backgammon playing (Qi and Sun, 2001) to automated generation of jazz melodies and solos (Papadopoulos, 1998; Biles, 1994). The task of GA is always similar: To find a minimum or maximum of the optimized function. The GA itself is a universal tool, while the most critical and application dependent stage of operation is the calculation of the fitness value. Care has to be taken when considering how to rank the candidate solutions.

This chapter concentrates on the fitness function that is used in conjunction with the GA. The aim of the fitness function is to calculate the perceptual similarity between two tones and give a grade for the tone that is under examination. A psychoacoustic model that accounts for the frequency masking behavior and frequency dependence of human hearing is designed.

4.1 Mean Squared Error of STFT Sequences

Human hearing analyzes sound both in the frequency and time domain. Since spectra of all musical sounds vary with time it is appropriate to calculate the spectral similarity in short time segments. A common method is to measure the mean squared error of the short-time spectra of the two sounds. The STFT of signal $y(n)$ is a sequence of discrete Fourier transforms (DFT) (Allen and Rabiner, 1977)

$$Y_m(k) = \sum_{n=0}^{N-1} w(n)y(n+mH)e^{-jw_k n}, \qquad m = 0, 1, 2, \ldots \tag{4.1}$$

with

$$w_k = \frac{2\pi k}{N}, \qquad k = 0, 1, 2, \ldots, N-1 \tag{4.2}$$

where N is the length of the DFT, $w(n)$ is a window function, and H is the hop size or time advance (in samples) per frame. When N is a power of two, e.g. 1024, each DFT can be computed efficiently with the FFT algorithm, which is the case in practise.

If $o(n)$ is the output sound of the synthesis model and $t(n)$ is the target sound then the mean squared error (MSE) (inverse of the fitness) of the candidate solution is calculated as follows

$$E = \frac{1}{F} = \frac{1}{L} \sum_{m=0}^{L-1} \sum_{k=0}^{N-1} (|O_m(k)| - |T_m(k)|)^2, \tag{4.3}$$

where $O_m(k)$ and $T_m(k)$ are the STFT sequences of $o(n)$ and $t(n)$ and L is the length of the sequences $m = 0, 1, 2, \ldots, L-1$. Normalized STFT sequences for two synthesized tones with different parameter values and the mean squared error surface are shown in Figure 4.1. The error value E is calculated by summing the frequency and time components of the surface, for this particular case $E = 21.2032$. Error values of various cases are not necessarily comparable because the fitness function is not an absolute metric but the values depend on the lengths of the analyzed signal and DFT. Also the coarse normalizing can affect values and a more sophisticated fitting technique for STFT sequences are required.

4.2 Frequency Masking

The analytical error calculated with Eq.(4.3) is a raw simplification from the viewpoint of auditory perception. Therefore, an auditory model is required. An important feature of human hearing is the frequency masking phenomenon, where a low-level signal, e.g, a pure tone (the maskee) can be made inaudible by simultaneously occuring stronger signal (the masker), e.g, narrow band noise, if masker and maskee are close enough

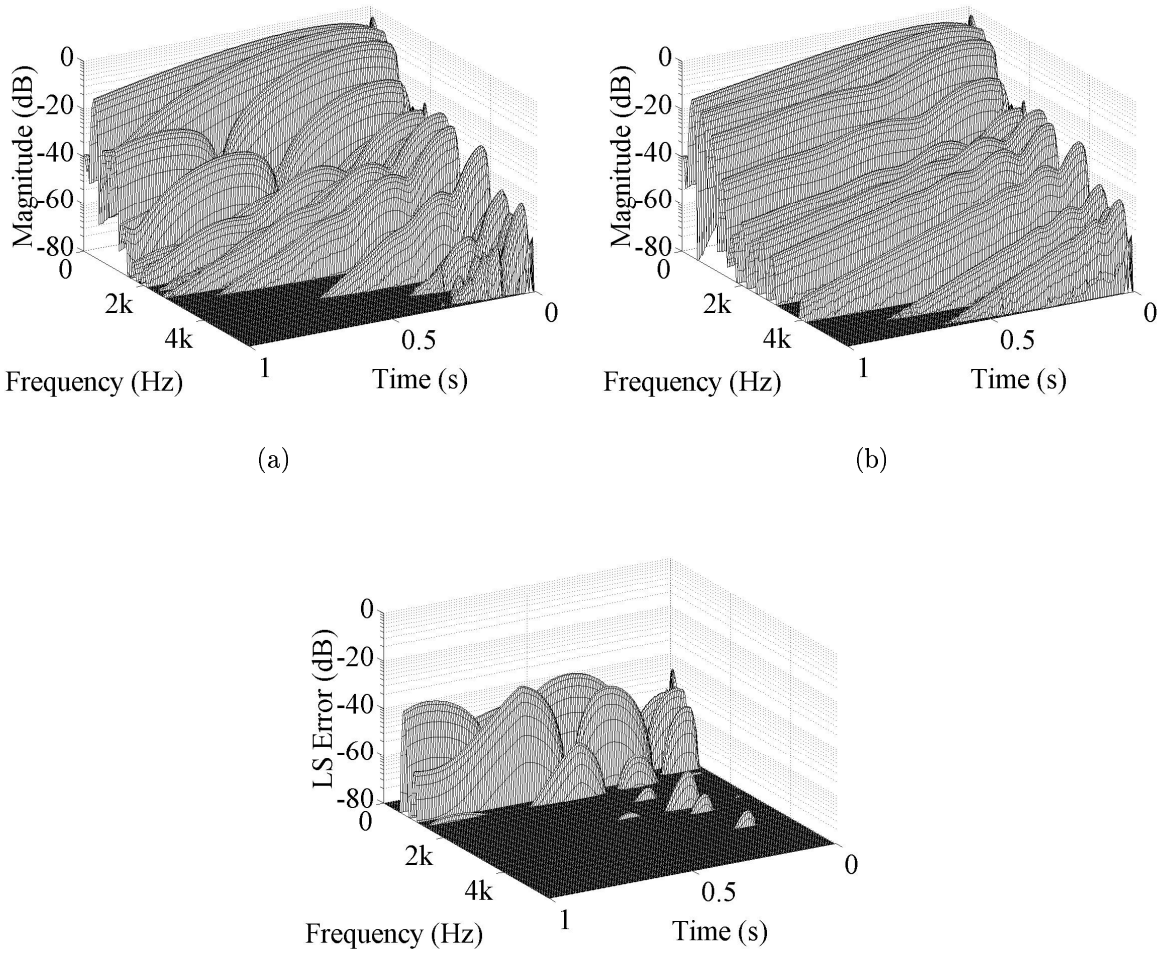


Figure 4.1: STFT sequences for two synthesized tones. (a) Parameter values: $f_{0,h} = 331$, $f_{0,h} = 331.5$, $g_h = 0.99$, $a_h = -0.1$, $g_v = 0.992$, $a_v = -0.2$, $m_p = 0.5$, $m_o = 0.5$, and $g_c = 0.1$. (b) Parameter values: $f_{0,h} = 331$, $f_{0,h} = 331.7$, $g_h = 0.985$,$a_h = -0.12$, $g_v = 0.994$, $a_v = -0.05$, $m_p = 0.2$, $m_o = 0.8$, and $g_c = 0.4$. (c) Mean squared error surface for the STFT sequences. The error value $E = 21.2032$.

to each other in frequency. The masking threshold depends on the sound pressure level (SPL), the frequency of the masker, and on the characteristics of the masker and maskee. The masking curves for narrow band noise when the center frequency of the masker is 250 Hz, 1 kHz or 4 kHz at the SPL of 60 dB are shown in Figure 4.2 (Zwicker and Zwicker, 1991). As can be seen the slopes are steeper towards lower frequencies, implying that the higher frequencies are more easily masked.

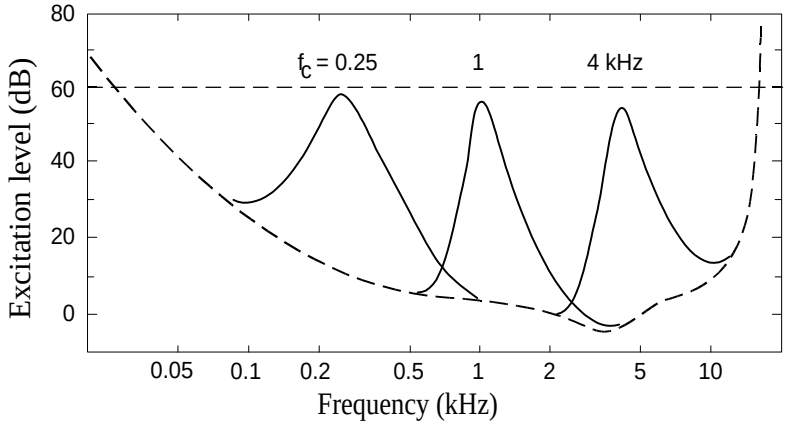


Figure 4.2: The masking curves for narrow band noise when the center frequency of the masker is 250 Hz, 1 kHz or 4 kHz at the SPL of 60 dB (Karjalainen, 1999).

One possibility to include the frequency masking properties to our model would be to measure the masking threshold for all partials and ignore the inaudible components in the error calculation. Such a method has been used to speed up additive synthesis (Lagrange and Marchand, 2001), data compression of sinusoidal modeling (Garcia and Pampin, 1999), and perceptual wavetable matching for synthesis of musical instrument tones (Wun and Horner, 2001; Wun et al., 2001). Partials are first tracked from each STFT spectrum and a simple masking model, as in Figure 4.3, is used to evaluate the signal-to-mask ratio (SMR) of each partial (Zwicker and Fastl, 1990). The model where the masking threshold is considered as a triangle in the Bark scale is an approximation and the behavior is not exactly realistic especially in the top of the triangle. The model consists of:

- The difference δ between the level of the masker and the masking threshold (typically -10 dB),
- The masking curve towards lower frequencies, or left slope (typically -27 dB/Bark),
- The masking curve towards higher frequencies, or right slope (typically -15 dB/Bark).

One disadvantage of the method is that it requires peak tracking of partials which is a time consuming procedure. In this thesis we use a technique which determines the

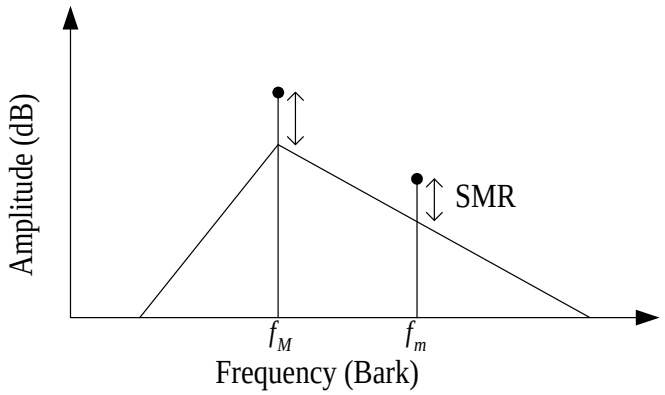


Figure 4.3: A sinusoid of frequency f_M masking another sinusoid of frequency f_m.

threshold of masking directly from the STFT sequences. The frequency components below that threshold are inaudible and therefore they are unnecessary when calculating the perceptual similarity. This technique proposed by Johnston, while at the Bell Labs, in (Johnston, 1988b) has been successfully included in an audio coder (Johnston, 1988a) and applied in perceptual error calculation (Garcia, 1998). The technique,also known as Johnston model, forms a basis to present day audio codecs, for example, the ISO/IEC MPEG-1 psychoacoustic model 2, which is often used in ".mp3" encoders, is a close relative to the Johnston's model (Painter and Spanias, 2000). Masking curve determination with MPEG-1, psychoacoustic model I (ISO/IEC 11172-3, 1993) is used for psychoacoustic modeling of audio in (Hermus et al., 2002). Improvement in the psychoacoustic model is related to the tonality estimation and prediction of individual frequency components.

4.3 Calculating the Threshold of Masking

The threshold of masking is calculated in several steps:

1. Windowing the signal and calculating STFT,
2. calculating the power spectrum for each DFT,
3. mapping the frequency scale into the Bark domain and calculating the energy per critical band,
4. applying the spreading function to the critical band energy spectrum,
5. calculating the spread masking threshold,

6. calculating the tonality dependant masking threshold,
7. normalizing the raw masking threshold and calculating the absolute threshold of masking.

First each DFT is converted to the power spectrum

$$P_m(k) = Re\{T_m(k)\}^2 + Im\{T_m(k)\}^2 = |T_m(k)|^2. \tag{4.4}$$

The frequency spectrum is then translated into the Bark scale by using the approximation (Zwicker and Fastl, 1990)

$$\nu = 13 \arctan\left(\frac{0.76f}{\text{kHz}}\right) + 3.5 \arctan\left(\frac{f}{7.5\text{kHz}}\right)^2, \tag{4.5}$$

where f is the frequency in Hertz and ν is the mapped frequency in Bark units. The energy in each critical band is the partial sum

$$Z_m(\nu) = \sum_{k=B_l(\nu)}^{B_h(\nu)} P_m(k), \qquad \nu = 1, 2, \ldots, N_c \tag{4.6}$$

where $B_h(\nu)$ is the upper boundary of the critical band ν, $B_l(\nu)$ is the lower boundary of the critical band ν, and N_c is the number of critical bands which depends on the sampling rate. N_c is 25 for the sample rate of 44.1 kHz. A power spectrum and energy per critical band for a 12 ms excerpt from a guitar tone are shown in Figure 4.4(a). The discrete representation of fixed critical bands is a close approximation and in reality each band builds up around a narrow band excitation. The effect of masking of each narrow band excitation spreads across all critical bands. This is described by a spreading function given in (Schroeder et al., 1979)

$$10 \log_{10} B(\nu) = 15.91 + 7.5(\nu + 0.474) - 17.5\sqrt{1 + (\nu + 0.474)^2}\text{dB}. \tag{4.7}$$

The spreading function is presented in Figure 4.5. The spreading effect is applied by convolving the critical band energy function with the spreading function (Johnston, 1988b)

$$S_{P,m}(\nu) = Z_m(\nu) * B(\nu), \tag{4.8}$$

where the asterisk corresponds to discrete convolution. Spread energy per critical band is shown in Figure 4.4(b).

Masking threshold depends on the characteristics of the masker and masked tone. Two different thresholds are detailed and used in (Johnston, 1988b). For the tone masking noise the threshold is estimated as $14.5 + \nu$ dB below the $S_{P,m}$. For noise masking the tone it is estimated as 5.5 dB below the $S_{P,m}$. Spectral flatness measure is used to determine the noiselike or tonelike characteristics of the masker. The spectral flatness measure V in decibels is defined as (Johnston, 1988b)

$$V_m = 10 \log_{10} \frac{\left[\prod_{k=0}^{N-1} P_m(k)\right]^{\frac{1}{N}}}{\frac{1}{N}\sum_{k=0}^{N-1} P_m(k)} \tag{4.9}$$

that is the ratio of the geometric to the arithmetic mean of the power spectrum. The tonality factor α is defined with the V_m as follows

$$\alpha_m = \min\left(\frac{V_m}{V_{max}}, 1\right), \tag{4.10}$$

where $V_{max} = -60$ dB. That is to say that if the masker signal is entirely tonelike $\alpha = 1$, and if the signal is pure noise $\alpha = 0$. The tonality factor is used to weight geometrically the two thresholds mentioned above to form the masking energy offset $U_m(\nu)$ for each band

$$U_m(\nu) = \alpha_m(14.5 + \nu) + (1 - \alpha_m)5.5. \tag{4.11}$$

The offset is then subtracted from the spread spectrum to estimate the raw masking threshold shown in Figure 4.4(c)

$$R_m(\nu) = 10^{\log_{10}(S_{P,m}(\nu)) - \frac{U_m(\nu)}{10}}. \tag{4.12}$$

Convolution of the spreading function and the critical band energy function increases the energy level in each band. Normalization procedure used in (Johnston, 1988b) takes this into account and divides each component of $R(\nu)$ by the number of points in the corresponding band

$$Q_m(\nu) = \frac{R_m(\nu)}{N_p(\nu)}, \tag{4.13}$$

where $N_p(\nu)$ is the number of points in each band ν. The final threshold of masking $W_m(k)$ is calculated by comparing the normalized threshold to the absolute threshold of hearing and mapping from Bark to the frequency scale. The most sensitive area in human hearing is around 4 kHz. If the normalized energy $Q_m(\nu)$ in any critical band is lower than the energy in a 4 kHz sinusoidal tone with one bit of dynamic range, it is changed to the absolute threshold of hearing. This is a simplified method, used in (Johnston, 1988b), to set the absolute levels and in reality the absolute threshold of hearing varies with frequency.

An example of the final threshold of masking is shown in Figure 4.4(d). It is seen that most of the high partials and the background noise at the high frequencies are below the threshold and thus inaudible.

This is the traditional method for evaluating the auditory masking threshold. More efficient methods have been reported in the literature as in (Mourjopoulos and Tsoukalas, 1992) where the calculation was performed with neural networks.

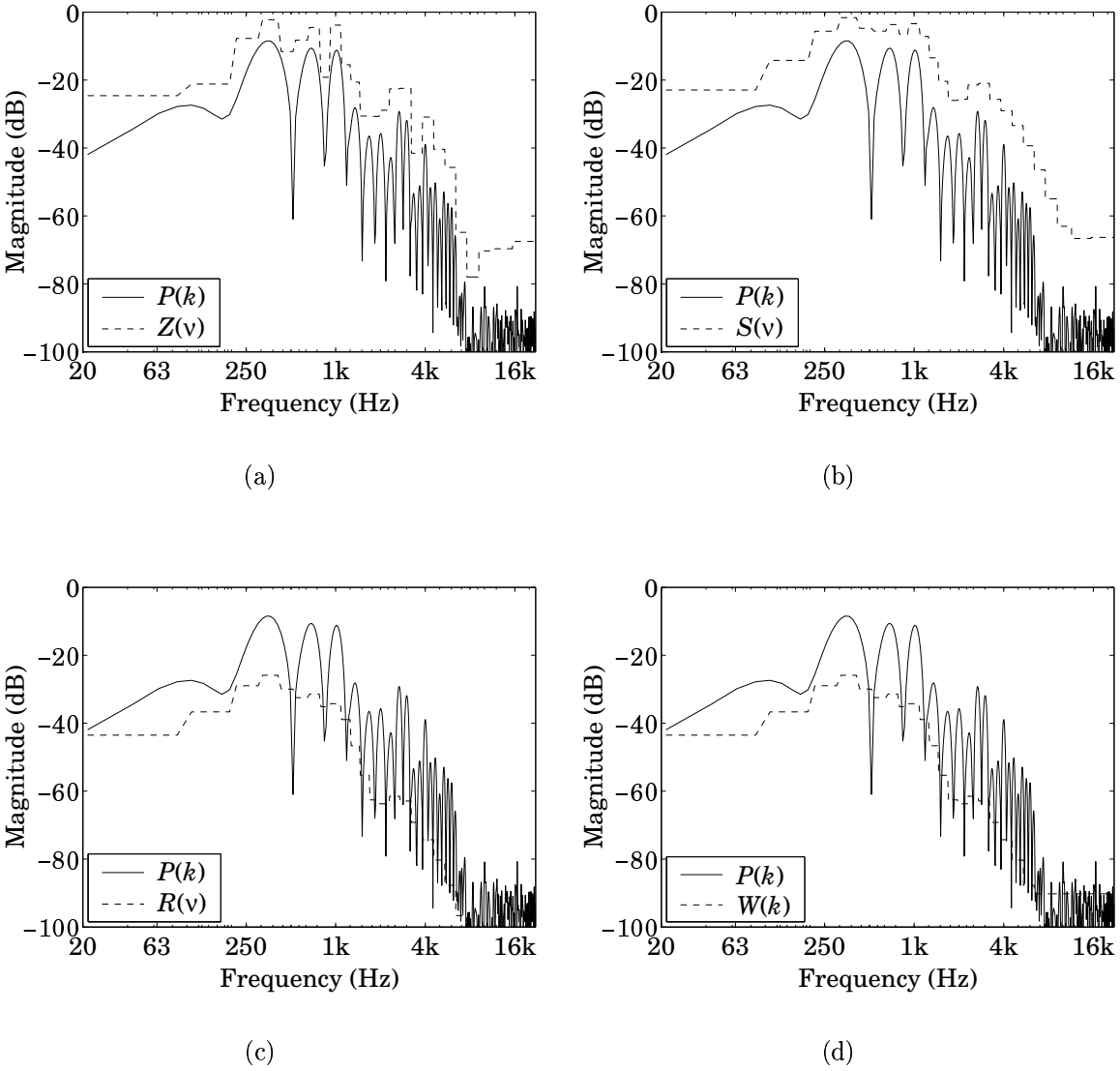


Figure 4.4: Determining the threshold of masking for a 12 ms excerpt from a recorded guitar tone. Fundamental frequency of the tone is 331 Hz. Power spectrum is shown with the solid line in Figures (a)-(d). Dashed line shows the calculated threshold. (a) Energy per critical band. (b) Spread energy per critical band. (c) Raw masking threshold. (d) Final masking threshold.

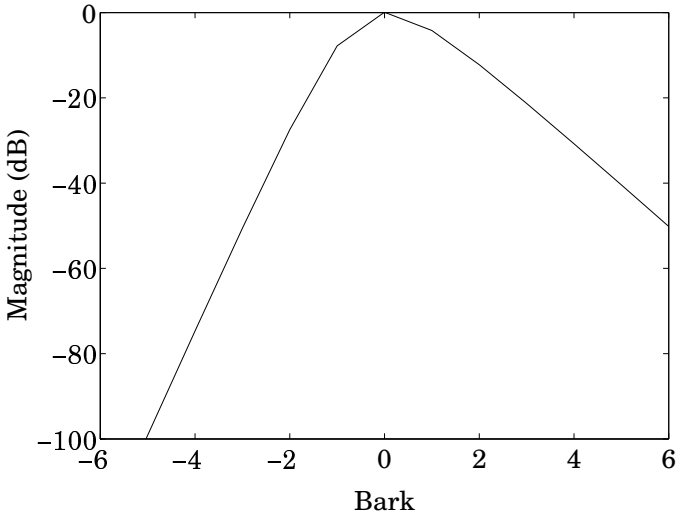


Figure 4.5: Spreading function.

4.4 Calculating the Perceptual Error

Perceptual error is calculated in (Garcia, 1998) by weighting the error from Eq. 4.3 with two matrices

$$W_{f,m}(k) = \begin{cases} 1 & \text{if } T_m(k) \geq W_m(k) \\ 0 & \text{otherwise} \end{cases} \tag{4.14}$$

$$W_{t,m}(k) = \begin{cases} 1 & \text{if } O_m(k) \geq W_m(k) \text{ and } T_m(k) < W_m(k) \\ 0 & \text{otherwise} \end{cases} \tag{4.15}$$

Matrices are defined such that the full error is calculated for spectral components which are audible in a recorded tone $T_m(k)$ (that is above the threshold of masking). Matrix $W_{f,m}(k)$ is used to account for these components. For the components which are inaudible in a recorded tone but audible in the sound output of the model $O_m(k)$ the error between the sound output and the threshold of masking is calculated. Matrix $W_{t,m}(k)$ is used to weight these components.

Perceptual error E_p is a sum of these two cases. No error is calculated for the components which are below the threshold of masking in both sounds. The error calculation is illustrated in Figure 4.6. The error is calculated according to the shaded area as follows

- Full error is calculated when $F_0 \leq f \leq F_1$ or $F_3 \leq f \leq F_4$, since $T(k) \geq W(k)$

- Partial error between the synthesized tone's spectrum and the threshold of masking is calculated when $F_1 < f \leq F_2$ or $F_4 < f \leq F_5$, since $O(k) \geq W_m(k)$ and $T_m(k) < W_m(k)$.
- No error is calculated when $F_2 < f < F_3$.

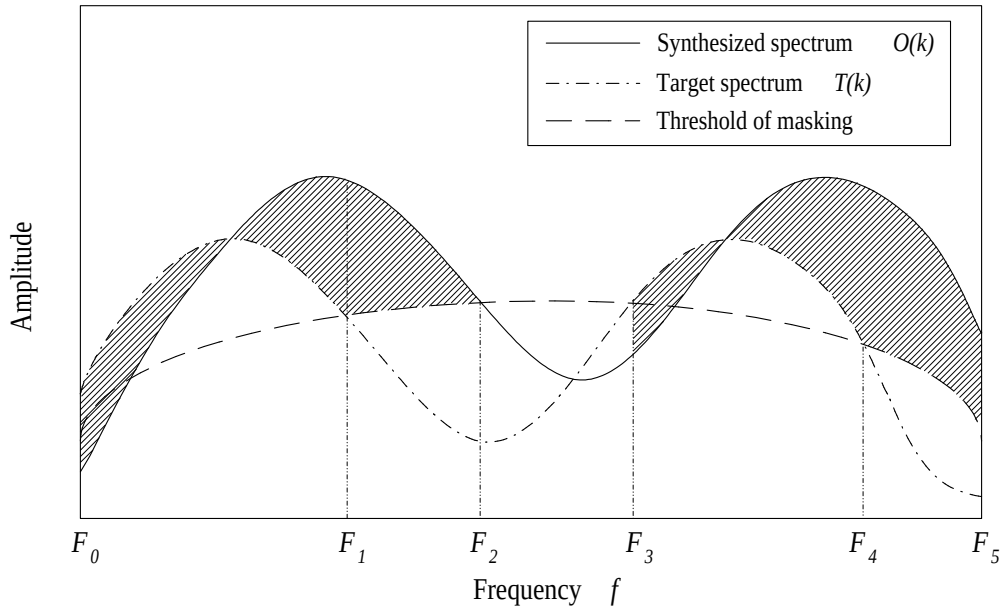


Figure 4.6: Error is calculated according to the shaded area. Full error is calculated when $F_0 \leq f \leq F_1$ or $F_3 \leq f \leq F_4$. Partial error is calculated when $F_1 < f \leq F_2$ or $F_4 < f \leq F_5$. No error is calculated when $F_2 < f < F_3$

The sensitivity of the ear varies with the frequency and the quality of the sound. This phenomena was examined by Fletcher and Munson, who first determined the equal loudness curves for pure tones (Fletcher and Munson, 1933). Our auditory model accounts for the frequency dependence of human hearing. This is done by weighting the error with an inverted equal loudness curve at sound pressure level of 60 dB shown in Figure 4.7. This is not exactly the case in practice, especially for the very quiet or loud tones, but it is reasonable for the average listening volumes.

The overall amplitude of STFTs $O_m(k)$ and $T_m(k)$ can vary depending on the excitation signal and the parameter values. However, the aim is to calculate dissimilarities in the shape of spectra and to find two similarly shaped STFT surfaces. Therefore, the amplitude of the spectrum of the synthesized tone has to be equalized with the target spectrum. This is done by multiplying $|O_m(k)|$ with the normalizing gain p which is defined as follow

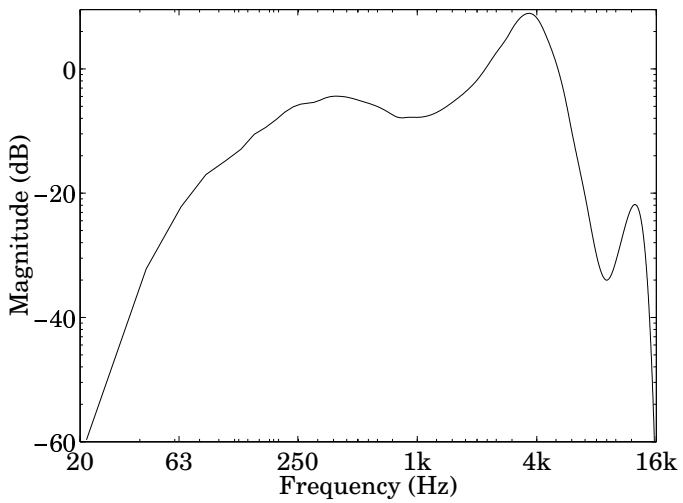


Figure 4.7: The frequency dependant weighting function, which is the inverse of the equal loudness curve at the SPL of 60 dB.

$$p = \sqrt{\frac{\sum_{k=0}^{N-1}\sum_{m=0}^{L-1}|T_m(k)|^2}{\sum_{k=0}^{N-1}\sum_{m=0}^{L-1}|O_m(k)|^2}} \tag{4.16}$$

Finally, the perceptual error function is evaluated as

$$\begin{aligned} E_p &= \frac{1}{F_p} = \frac{1}{L}\sum_{k=0}^{N-1} W_s(k) \sum_{m=0}^{L-1} \{[(p|O_m(k)| - |T_m(k)|)^2 W_{fm}(k)] + \cdots \\ &\quad \cdots + [(p|O_m(k)| - |T_m(k)|)^2 W_{tm}(k)]\}, \end{aligned} \tag{4.17}$$

where $W_s(k)$ is the inverted equal loudness curve used to imitate the frequency dependence of human hearing.

Chapter 5

Implementation

Theory of the synthesis model, genetic algorithms, and fitness calculation are explained in previous chapters. Operation of genetic algorithms is explained in Chapter 3 and the fitness function that is used in this thesis is described in Chapter 4. This chapter is about implementation of the parameter estimation procedure that is founded on the theory.

At the beginning of this chapter the parameter space is discretized to reduce the number of possible parameter sets for the model. Results from previous studies are used as a basis for discretization, but the exact discrimination levels for all parameters have not been proposed in literature, and therefore the sensitivity of parameters, examined with the fitness calculation method described in previous chapter, is used as a guideline to the discretization. Combining the results discrete grids are defined for nine parameters that are used to control the plucked string synthesis model. In the end of the chapter the genetic algorithm used for parameter estimation, side effects due to discretization, and modifications for standard GA operations are explained.

5.1 Discretizing the Parameter Space

The number of data points in the parameter space can be reduced by discretizing the individual parameters in a perceptually reasonable manner. Range of parameters can be reduced to cover only all the possible musical tones and deviation steps can be kept just below the discrimination threshold.

5.1.1 Decay Parameters

The audibility of variations in decay of the single string model in Figure 2.2 have been studied in (Järveläinen and Tolonen, 2001). Time constant τ of the overall decay was used to describe the loop gain parameter g while the frequency dependent decay was controlled directly by parameter a. Values of τ and a were varied and relatively large deviations in parameters were claimed to be inaudible. Järveläinen and Tolonen proposed that a variation of the time constant between 75% and 140% of the reference value can be allowed in most cases. An inaudible variation for the parameter a was between 83% and 116% of the reference value.

The discrimination thresholds were determined with two different tone durations 0.6 s. and 2.0 s. In our study the judgement of similarity between two tones is done by comparing the entire signals and therefore the results from (Järveläinen and Tolonen, 2001) cannot be directly used for parametrization of a and g. The tolerances are slightly smaller because the judgement is made based on not only the decay but also the duration of a tone. Based on our informal listening test and including a margin of certainty we have defined the variation to be 10% for the τ and 7% for the parameter a. The parameters are bounded so that all the playable musical sounds from tightly damped picks to very slowly decaying notes are possible to produce with the model. This results in 62 discrete non-uniformly distributed values for g and 75 values for a as shown in Figures 5.1(a) and 5.1(b). The corresponding amplitude envelopes of tones with different g parameter are shown in Figure 5.1(c). Loop filter magnitude responses for varying parameter a with $g = 1$ are shown in Figure 5.1(d).

5.1.2 Fundamental Frequency and Beating Parameters

The fundamental frequency estimate $\hat{f}_0$ from the calibrator is used as an initial value for both polarizations. When the fundamental frequencies of two polarizations differ the frequency estimate settles in the middle of the frequencies as shown in Figure 5.2. Frequency discrimination thresholds as a function of frequency have been proposed in (Wier et al., 1977). Also the audibility of beating and amplitude modulation has been studied in (Zwicker and Fastl, 1990). These results do not give us directly the discrimination thresholds for the difference in the fundamental frequencies of the two polarization string model, because the fluctuation strength in a output sound depends on the fundamental frequencies and the decay parameters g and a.

The sensitivity of parameters can be examined when a synthesized tone with known parameter values is used as a target tone with which another synthesized tone is compared. Varying one parameter after another and freezing the others we obtain the error

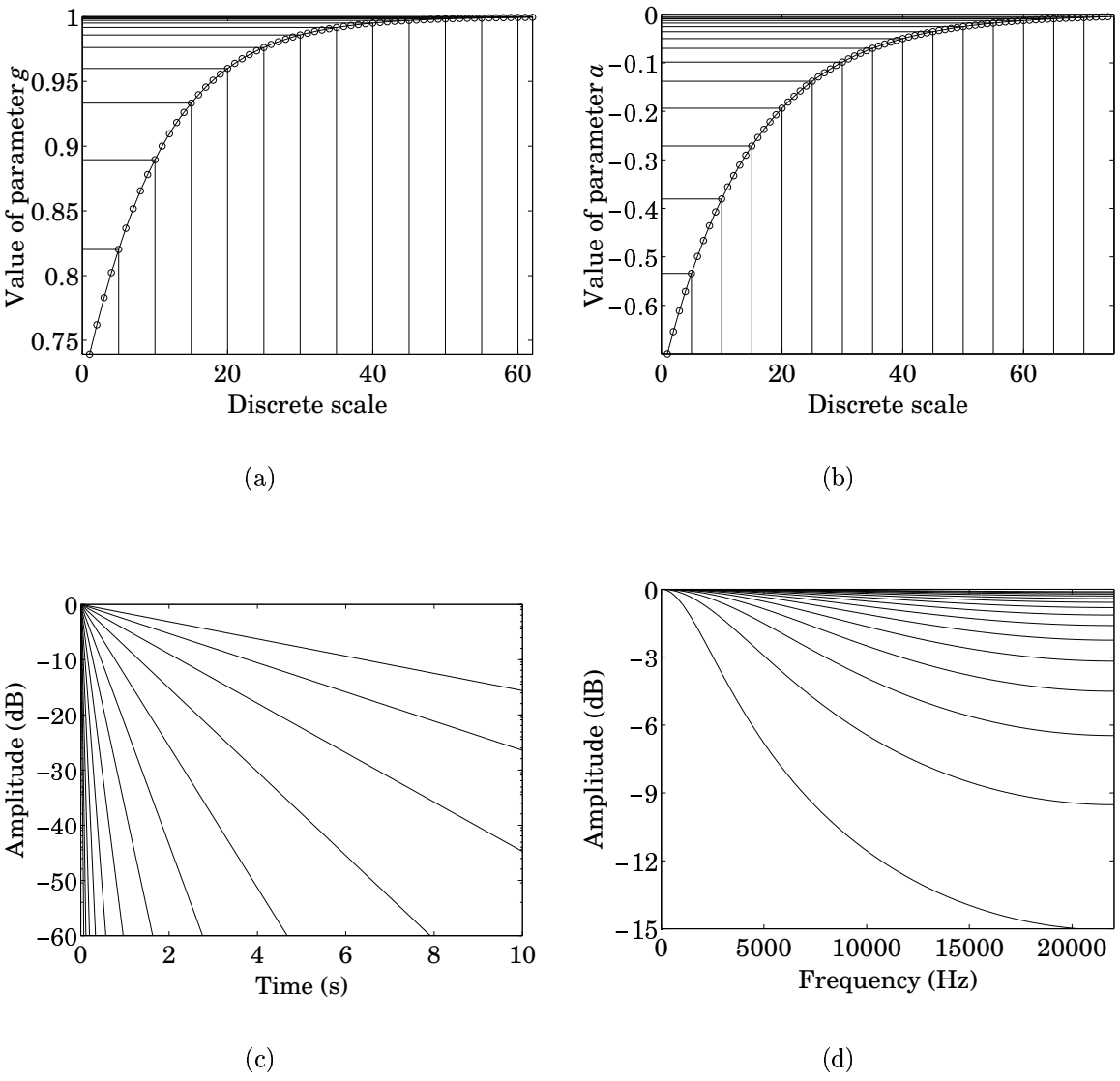


Figure 5.1: Discretizing the parameters g and a. (a) Discrete values for the parameter g when $f_0 = 331$ and the variation for the time constant τ is 10%. (b) Discrete values for the parameter a when the variation is 7%. (c) Amplitude envelopes of tones with different discrete values of g. (d) Loop filter magnitude responses for different discrete values of a when $g = 1$.

as a function of the parameters. In Figure 5.3(a) the target values of $f_{0,v}$ and $f_{0,h}$ are 331 and 330 Hz. The solid line shows the error when $f_{0,v}$ is linearly swept from 327 to 344 Hz. The global minimum is obviously found when $f_{0,v} = 331$ Hz. Interestingly another non-zero local minimum is found when $f_{0,v} = 329$ Hz that is when the beating is similar. The dashed line shows the error when both $f_{0,v}$ and $f_{0,h}$ are varied but the difference in the fundamental frequencies is kept constant. It can be seen that the difference is more dominant than the absolute frequency value and have to be therefore discretized with higher resolution. Instead of operating the fundamental frequency parameters directly we optimize the difference $d_f = |f_{0,v} - f_{0,h}|$ and the mean frequency

$f_0' = |f_{0,v} + f_{0,h}|/2$ individually. Combining previous results from (Wier et al., 1977) and (Zwicker and Fastl, 1990) with our informal listening test we have discretized d_f with 100 and f_0' with 20 discrete values. The range of variation is set as follows

$$r_p = \pm\left(\frac{\hat{f}_0}{10}\right)^{\frac{1}{3}}, \tag{5.1}$$

which is shown in Figure 5.3(b).

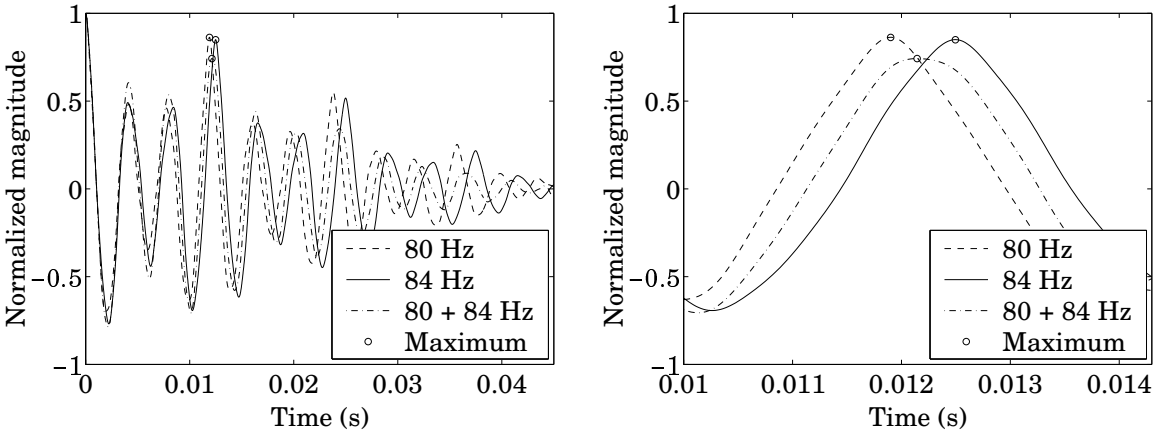
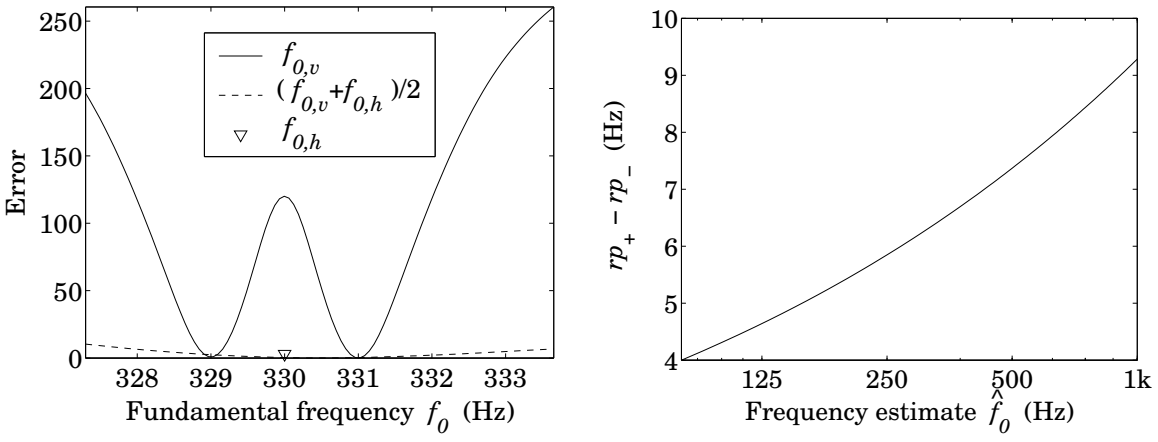


Figure 5.2: Three autocorrelation functions. Dashed and solid lines show functions for two single polarization guitar tones with fundamental frequencies of 80 Hz and 84 Hz. Dash-dotted line corresponds to a dual-polarization guitar tone with fundamental frequencies of 80 Hz and 84 Hz. (a) Entire autocorrelation function. (b) Zoomed around the maximum.

5.1.3 Other Parameters

The tolerances for the mixing coefficients m_p, m_o, and g_c have not been studied and the parameters have been earlier adjusted by trial and error (Laurson et al., 2001). Therefore, no initial guesses are made for these parameters. The sensitivities for the mixing coefficients are examined in an example case in Figure 5.4(a) where $m_p = 0.5$ and $m_p = 0.5$. It can be seen that the parameters m_p and m_o are most sensitive near the boundaries. The range for m_p and m_o are discretized according to the hyperbolic tangent between the range $x = [-2, 2]$ as follows



(a) (b)

Figure 5.3: (a) Error as a function of the fundamental frequencies. The target values of $f_{0,v}$ and $f_{0,h}$ are 331 Hz and 330 Hz. The solid line shows the error when $f_{0,h} = 330$ and $f_{0,v}$ is linearly swept from 327 to 334 Hz. The dashed line shows the error when both frequencies are varied simultaneously while the difference remains similar. (b) Range of variation in fundamental frequency as a function of frequency estimate from 80 Hz to 1000 Hz.

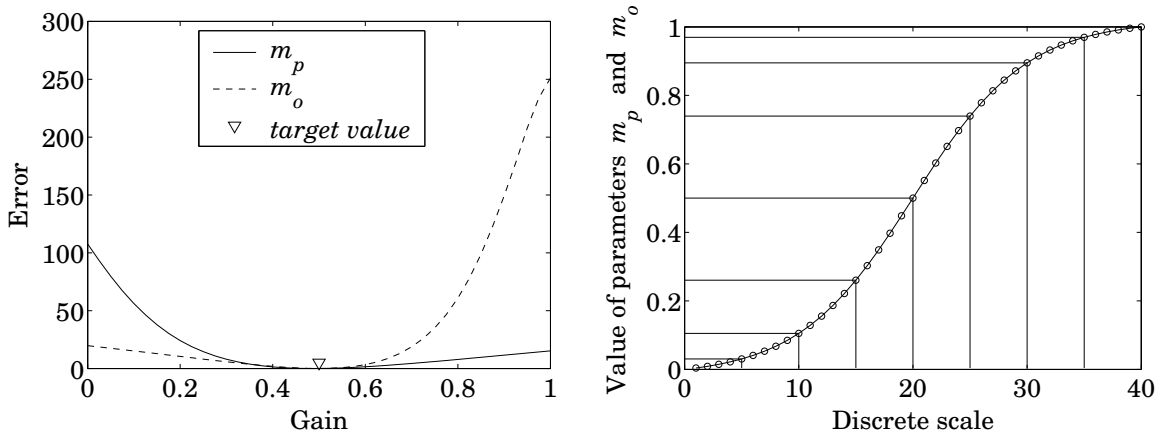
$$\frac{\tanh(x) + \tanh(2)}{2\tanh(2)}, \qquad x = [2, 2] \tag{5.2}$$

where the range $x = [-2, 2]$ is uniformly sampled with 40 values. The sampling grid is shown in Figure 5.5(b). The sensitivity for the parameter g_c is presented in Figure 5.4(a) where the target value $g_c = 0.1$. To obtain denser distribution near the zero we use the non-uniform sampling grid with 40 values in Figure 5.5 for the parameter g_c the range of which is limited to 0-0.5.

Discretizing the nine parameters this way results in $2.77 \cdot 10^{15}$ combinations in total for a single tone. For an acoustic guitar about 120 tones with different dynamic levels and playing styles have to be analyzed. It is obvious that an exhaustive search is impracticable.

5.1.4 Sensitivity of the Fitness Function

Error sensitivities of the mean fundamental frequency f_0' and the difference of the fundamental frequencies d_f are shown in Figure 5.3(a). Sensitivities of m_p, m_o, and g_c



(a) (b)

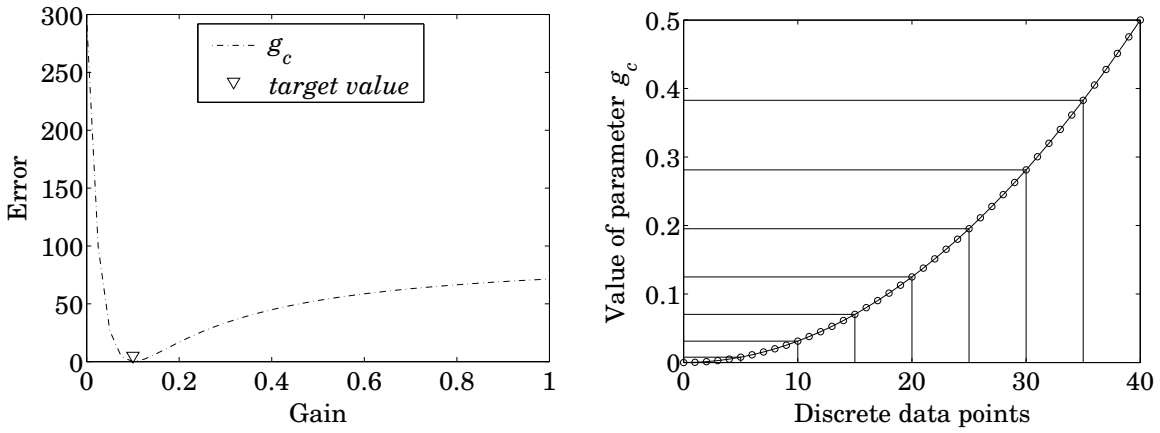
Figure 5.4: (a) Error as a function of the mixing coefficients m_p, m_o. The target values $m_p = m_o = 0.5$. (b) Discrete values for the parameters m_p and m_o.

are presented in Figures 5.4(a) and 5.5(a). As can be seen the error curves for d_f, m_p, m_o, and g_c are steep and the optimum can be assumed to be sensitive. The maximum error varies from 100 to 250, which is noticeably larger than with f_0'. Error Sensitivities of parameters g and a are shown in Figure 5.6. The error curves of the parameter g seems to be even more steeper than the other parameters, implying g to be the most sensitive parameter. In contrast, the parameter a is noticed to be ten times less sensitive that the parameter g. The maximum error is about 20, while the maximum error of g is more than 200. Behavior of error is similar although the fundamental frequency is different. Difference in sensitivity of parameters does not necessarily imply problems but supposedly the more sensitive parameters converge first while the less sensitive parameters settle in the later stage of the algorithm. The algorithm is not as stabile as it could be.

5.2 The Algorithm

A floating point implementation of GA is used with the parameter estimation procedure in this thesis. The algorithm is implemented as follows:

0. Analyze the recorded tone to be resynthesized using the calibration methods discussed in Section 2.2.1. The range of the parameter f_0' is chosen and the excitation signal is produced according to these results. Calculate the threshold of masking (Section 4.3) and the discrete scales for the parameters (Section 5.1).



(a) (b)

Figure 5.5: (a) Error as a function of the coupling coefficient g_c. The target value $g_c = 0.1$. (b) Discrete values for the parameter g_c.

1. Initialization: Create a population of S_p individuals (chromosomes). Each Chromosome is represented as a vector array $\vec{x}$ with nine components (genes) which contains the actual parameters. The initial parameter values are randomly assigned.

2. Fitness calculation: Calculate the perceptual fitness for each individual in the initial population.

   Repeat the following steps until termination

3. Selection of individuals: Select individuals for the mating pool by the normalized geometric ranking scheme (Section 3.3).

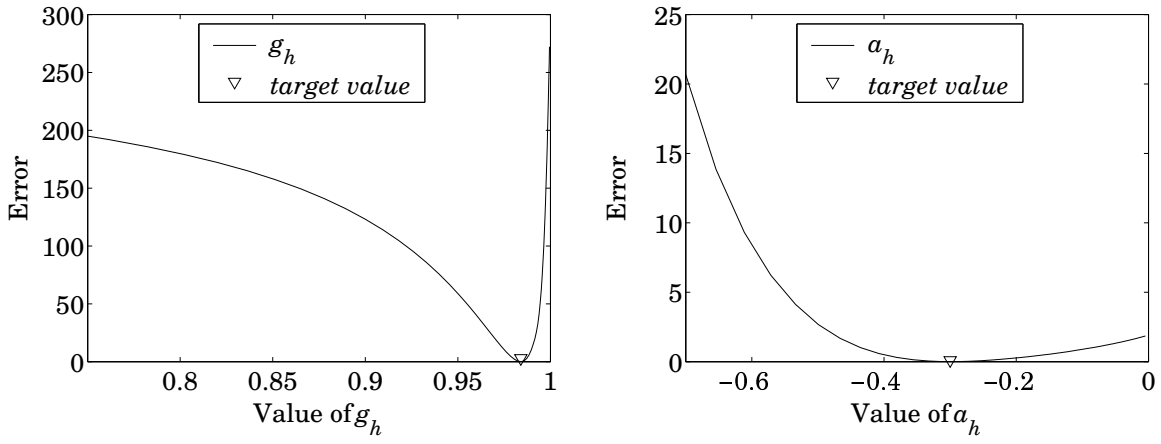
4. Crossover: Pick randomly a specified number of parents from selected individuals. Offspring is produced by crossing the parents with the simple, arithmetical, and heuristic crossover schemes (Section 3.4.1).

5. Mutation: Pick randomly a specified number of individuals for mutation. Uniform, non-uniform, multi-non-uniform, and boundary mutation schemes are used (Section 3.4.2).

6. Evaluate: Calculate the perceptual fitness for each new individual.

7. Replace the current population with the new one.

Floating point representation is used due to its fast operation compared to the binary based algorithm. Discrete parameter space and floating point numbers might appear to



(a) (b)

Figure 5.6: (a) Error as a function of the loop gain g_h. The target value $g_h = 0.984$ (b) Error as a function of the frequency dependent loop gain a_h. The target value $a_h = -0.3$. $f_0' = 330.5$ in both examples.

be a inconsistent combination, because an advantage of floating point representation is the higher precision of the algorithm. In practise, the precision of an algorithm depends on the precision used for floating point numbers. However, the algorithm has proven workable and results have been consistent. Few modifications to the original operators have been made.

All crossover schemes function with the discrete grid. Simple, arithmetical, and heuristic schemes produce offspring as originally. The only differences compared to true floating point numbers are that if single parameter values in parents are inside one discrete step the GA crosses the values over and over without a change in the error value. This is not a real problem because sooner or later single parameter values will converge to a single value among a population.

Uniform and boundary mutations operate without modifications. Uniform mutation operates uniformly over the space regardless of the ration G/N_g, which is the ratio of the current generation to the maximum number of generations. Non-uniform mutation with original definition becomes inoperative when G/N_g approaches number one. At the later stage of the algorithm average values for $f(G)$ are too small to chance the parameter values in a discrete grid. Therefore, the minimum value for $f(G)$ is defined to be 0.05 implying that if the parameter i has the twentieth value in our 40 point discrete scale the parameter is mutated at least one step. This can be seen in Figure 5.7, which is a modified case of Figure 3.4 in Chapter 3. The curves are forced not to get values below 0.05.

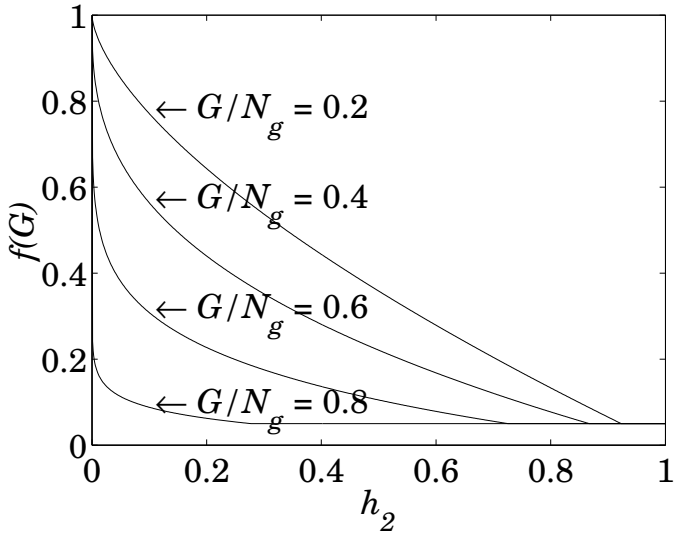


Figure 5.7: $f(G)$ for four selected moments while $b = 2$. Minimum value for $f(G)$ is 0.05.

The boundary mutation sets a parameter to one of its boundaries. It is useful if the optimal solution is supposed to lie near the boundaries of the parameter space. The scheme is used in special cases such as with staccato tones. A random number is used for the arithmetical crossover parameter c.

5.3 STFT

A special *Pitch synchronous Fourier transform* scheme, where the window length L_w is synchronized with the period length of the signal such that $L_w = 4f_s/f_0$ is utilized in this thesis. The used Hanning window for L_w points is defined as

$$w(n) = 0.5\Big(1 - \cos\Big(2\pi\frac{n+1}{N+1}\Big)\Big) \qquad n = 0, 1, 2, \ldots, L_w - 1 \tag{5.3}$$

The overlap of the windows is 50%, implying that $H = L_w/2$. Our sampling rate $f_s = 44100$ Hz by default and the length of FFT $N = 2048$. If the window function has less than N points the signal buffer is padded with zeros to reach the $N = 2048$. Care have to taken when the window function has more than N points, in which case the windowed signal is truncated. $L_w > N$ when $f_0 < 86.13Hz$ which is valid for the lowest guitar note. If $f_0' < 90Hz$ FFT length is changed to $N = 4096$.

5.4 Implementation

The block diagram of the parameter estimation scheme is presented in Figure 5.8. The implementation of the genetic algorithm is based on the Matlab toolbox *The Genetic Algorithm Optimization Toolbox (GAOT) for Matlab 5* developed by Houck et al. (1995), in North Carolina State University. The toolbox is available in the World Wide Web at `http://www.ie.ncsu.edu/mirage/GAToolBox/gaot/`. M-files for fitness function calculation, non-uniform, multi-non-uniform mutation, and file handling properties has been implemented for the GAOT-toolbox, the original routines of which have been modified slightly.

The fundamental frequency of a recorded target tone is first analyzed in pitch estimation block. The frequency estimate is used to calculate the pitch synchronous STFT and discrete parameter scales. Originally the pitch estimation and STFT-analysis blocks were built into the Calibrator, but since the frequency estimate and STFT are required also elsewhere the operations are performed off-line. Short time spectra are used to calculate the threshold of masking and the excitation signal is extracted from the target tone by the Calibrator. Analyzed data are fed into the genetic algorithm, which estimates a set of parameters.

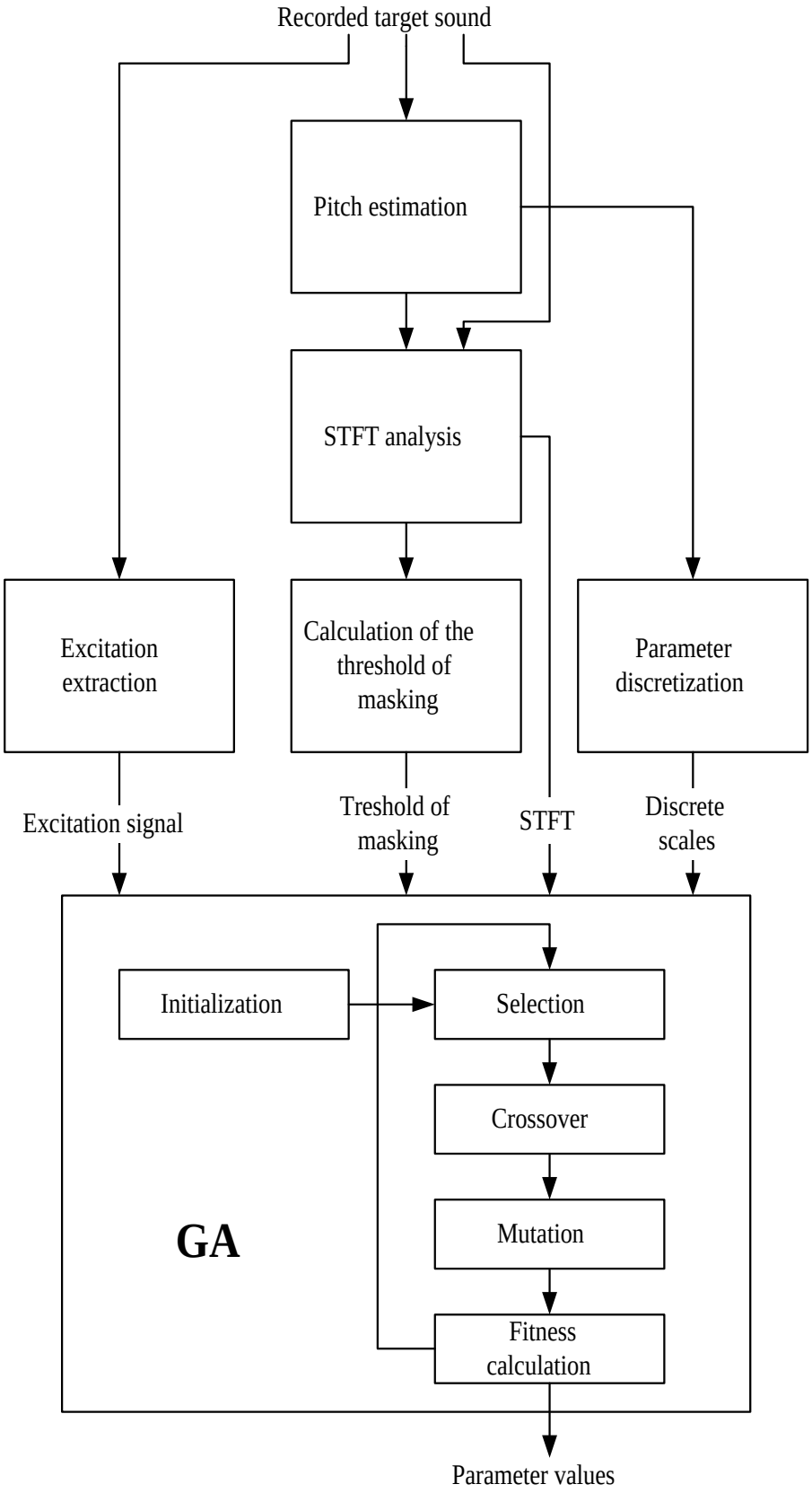


Figure 5.8: Block diagram of the parameter estimation procedure

Chapter 6

Experimentation and Results

In this chapter the efficiency of the proposed method is studied. First the parameters are estimated for the sound produced by the synthesis model itself. The advantage of using a synthetic signal as a target tone is the possibility to use predefined parameters. If the parameters are set according to the discrete grid there will always be a solution that gives the zero error and exactly reproduces the target tone.

First the same excitation signal extracted from a recorded tone by the method described in (Välimäki and Tolonen, 1998) was used for target and output sounds. A more realistic case is simulated when the excitation for resynthesis is extracted from the target sound. Finally, the model was tested with real recorded tones of different decaying and dynamic characteristics.

The system was implemented with Matlab software for the Microsoft Windows 98 operating system. All runs were performed on an Intel Pentium III computer 833 Hz with 256 MB of RAM.

6.1 Synthesized Target Tone with an Original Excitation

Same excitation is used for the target and the synthesized tone in the first experiment. The parameters for the experiment 1 is set as follows:

> Population size $S_p = 60$, number of generations = 400, probability of selecting the best individual $q = 0.02$, degree of non-uniformity $b = 3$, retries

$w = 3$, number of crossovers = 18, Number of mutations = 18.

The original and the estimated parameters for three runs are shown in Table 6.1. The execution time for a run was about 5 hours. The exact parameter values are estimated for the difference d_f and for the g_h and g_v in every run and for the a_v in run 1 and run 2. As noted in chapter 2 parameter values can be swapped between two polarizations if $m_p = 1 - m_o$, which is valid for our target values. Therefore, estimated decay parameters are swapped in run 3 and we get perfect agreement for parameters g_h, g_v, and a_v. Adjacent point in discrete grid is estimated for parameter a_h in every run and for the mean frequency f_0' in run three. As can be seen in Figure 5.3(a) the sensitivity for the mean frequency is negligible compared to the difference d_f, which might be the cause of deviations in mean frequency.

parameter	original parameter	estimated parameter (run1)	estimated parameter (run2)	estimated parameter (run3)
f_0'	330.5409	331.000850	330.000850	330.6799
d_f	0.8987	0.8987	0.8987	0.8987
g_h	0.9873	0.9873	0.9873	0.9873
a_h	-0.2905	-0.3108	-0.3108	-0.3108
g_v	0.9907	0.9907	0.9907	0.9907
a_v	-0.1936	-0.1936	-0.1936	-0.1477
m_p	0.5	0.2603	0.6024	0.3489
m_o	0.5	0.6971	0.3489	0.4483
g_c	0.1013	0.2628	0.0612	0.1378
error	–	0.0464	0.0465	0.0112

Table 6.1: Experiment 1. Original and estimated parameters when a synthesized tone with known parameter values are used as a target tone. The original excitation is used for the resynthesis.

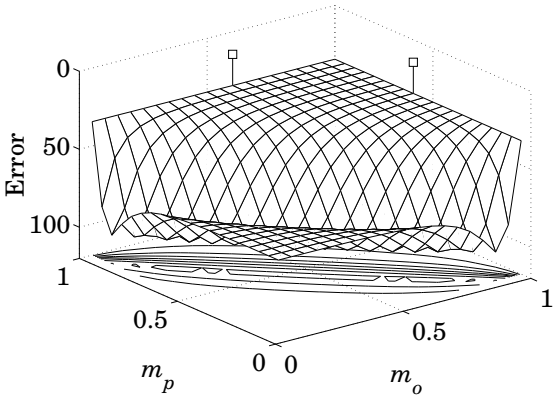
A conspicuous feature is the difference in the mixing parameters m_o, m_p and the coupling coefficient g_c. When running the algorithm multiple times no explicit optima for mixing and coupling parameters were found. However, synthesized tones produced by corresponding parameter values are indistinguishable. The behavior of two parameters can be studied by freezing the others and mapping all the points of the space of the two parameters into the related space of the error values. The resulting surface is called fitness or error landscape (Press, 1992). Error landscapes for mixing parameters with different values of g_c are shown in Figure 6.1 where the z-axis is reversed, implying that the minimum error is found in the highest point of an error landscape. Every other value in the nonlinear grid in Figure 5.5(b) is plotted for mixing parameters, which means that the actual minimum of an error landscape is not necessarily drawn in Figure. However, the shape of drawn landscapes are similar with real ones.

When the coupling path in the model is totally closed ($g_c = 0$) the minimum error value $E = 34.4081$ in Figure 6.1(a) is significant. Obviously, when $g_c = 0.08$, which is the target value, the minimum error is exactly zero as can be seen in Figure 6.1(d). Interestingly, the smallest error value, found in the intersecting point of two hogbacks, is very close to zero also in the other cases. In practice, the non-zero error value could be the consequence of the discrete sampling grid. The zero error might be found in every case if floating point numbers with sufficient precision were used.

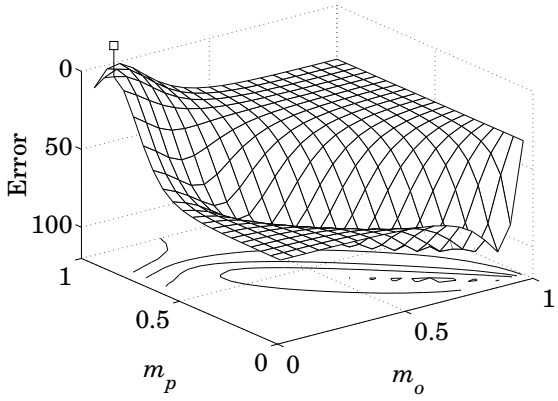
The single requirement for the negligible error seems to be the non-closed coupling path ($g_c > 0$). Another feature is that the error minimum is shown as a smooth hogback that rides along an error landscape while g_c is varied. This means that almost similar sounds which are very close to the optimal can be synthesized with various mixing parameters when g_c is fixed. A characteristic that can been noticed in runs one and two is that almost identical error value is obtained with parameters $m_p = 0.2603, m_o = 0.6971, g_c = 0.2628$ and $m_p = 0.6024, m_o = 0.3489, g_c = 0.0612$. With these values we get quite similar results for coefficients M_1 and M_2 at Equation 2.5 but especially the values of M_3 are very close to each other. $M_3 = 0.028706$ in the first and $M_3 = 0.028338$ in the second run. This kind of behavior has been noticed throughout the experiments, although the run three, which gives the best error value does not follow the trend. Looking Figure 6.1 verifies the conclusion. When g_c is increased the optimal value of m_p is decreasing and m_o is increasing.

However, it is obvious that the parameters m_p, m_o and g_c are not orthogonal which is clearly a problem with the model and also impairs efficiency of our parameter estimation algorithm. The implementation of the coupling effect and beating should be reconsidered in future.

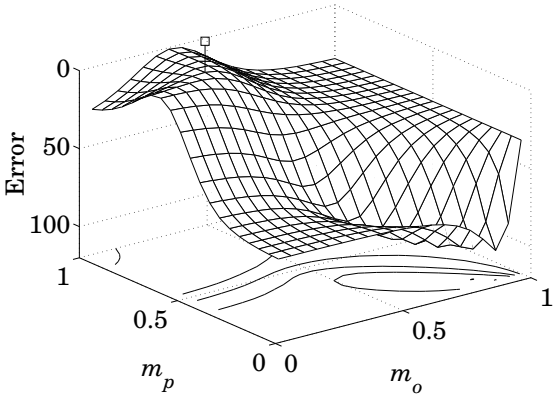
Rather than look into the exact parameter values it is better to analyze the quality of the tones produced with the parameters. In Figures 6.2 and 6.3 time domain envelopes and the eight first partials for the synthesized target tone with known parameter values and for the synthesized tones that uses estimated parameter values according to Table 6.1 are presented. As can be seen the envelopes are exactly similar and the partial envelopes match well. Only negligible dissimilarities can be noticed in beating amplitude in time domain, where the small dip in approximately 0.3 seconds from beginning of the tone is slightly deeper in estimated tones. According to our informal listening the variations in tones are inaudible.



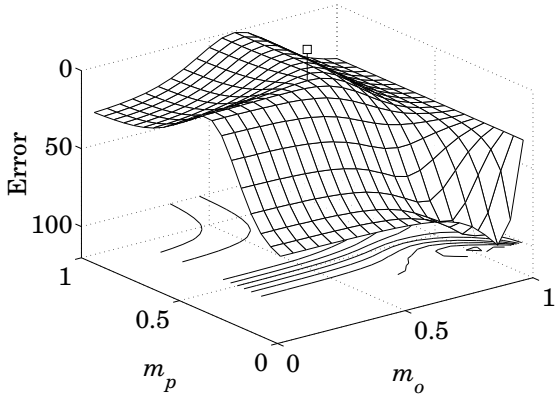
(a) $g_c = 0$, minimum error $E = 34.4081$ at $m_p = 1$ and $m_o = 0.6971$ or $m_p = 0.6971$ and $m_o = 1$.



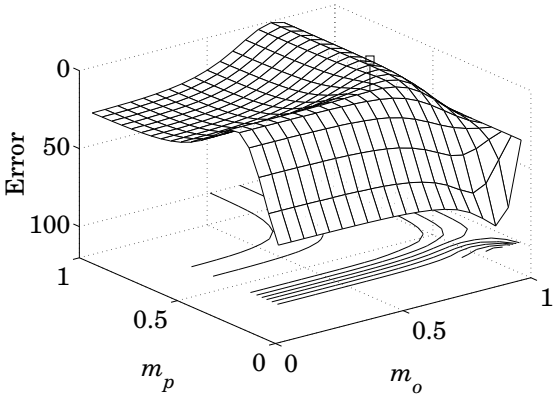
(b) $g_c = 0.0013$, minimum error $E = 0.0933$ at $m_p = 0.9780$ and $m_o = 0.0089$.



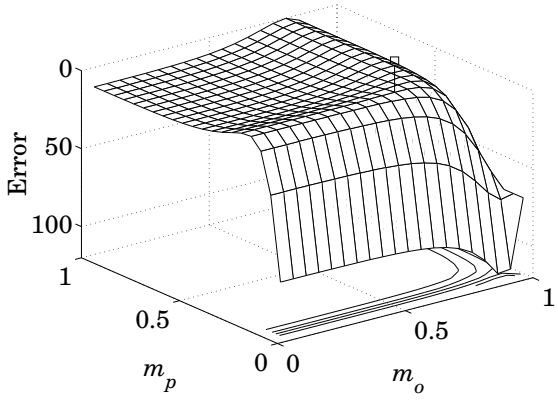
(c) $g_c = 0.0113$, minimum error $E = 0.0592$ at $m_p = 0.8950$ and $m_o = 0.1556$.



(d) $g_c = 0.0800$, minimum error $E = 0$ at $m_p = 0.5$ and $m_o = 0.5$.



(e) $g_c = 0.3200$, minimum error $E = 0.0436$ at $m_p = 0.1556$ and $m_o = 0.6971$.



(f) $g_c = 1$, minimum error $E = 0.0090$ at $m_p = 0.1050$ and $m_o = 0.7785$.

Figure 6.1: Error landscapes for parameters m_p and m_o with different values of g_c. Z-axis is reversed and the grid for mixing parameters accords to the discretizing scheme in Figure 5.5(b). Target values $m_p = 0.5$, $m_o = 0.5$, and $g_c = 0.08$. Other parameters $f_0' = 330.5$, $d_f = 0.8986$, $g_h = 0.9925$, $a_h = -0.2071$, $g_v = 0.9873$, $a_v = -0.2715$.

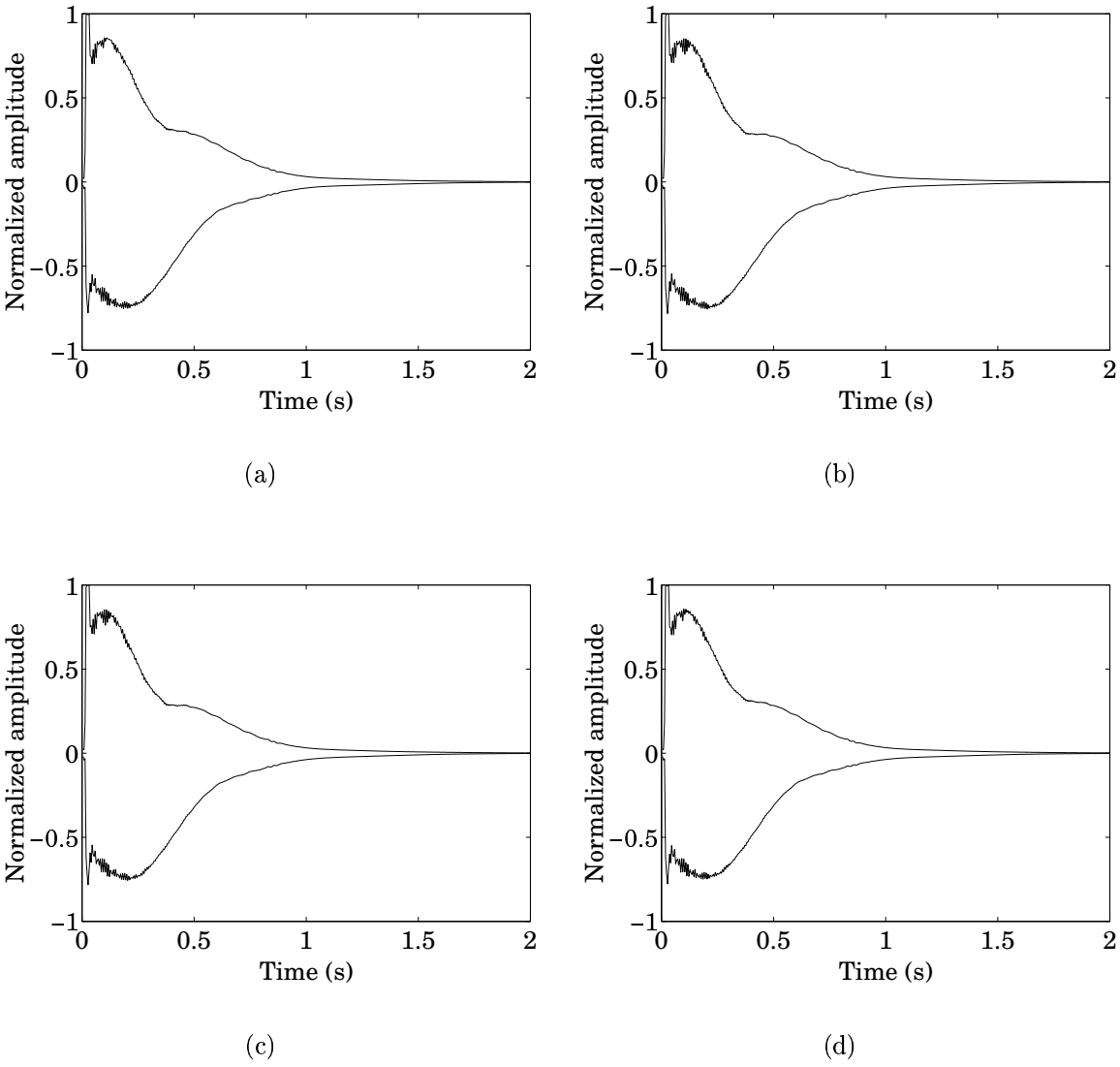
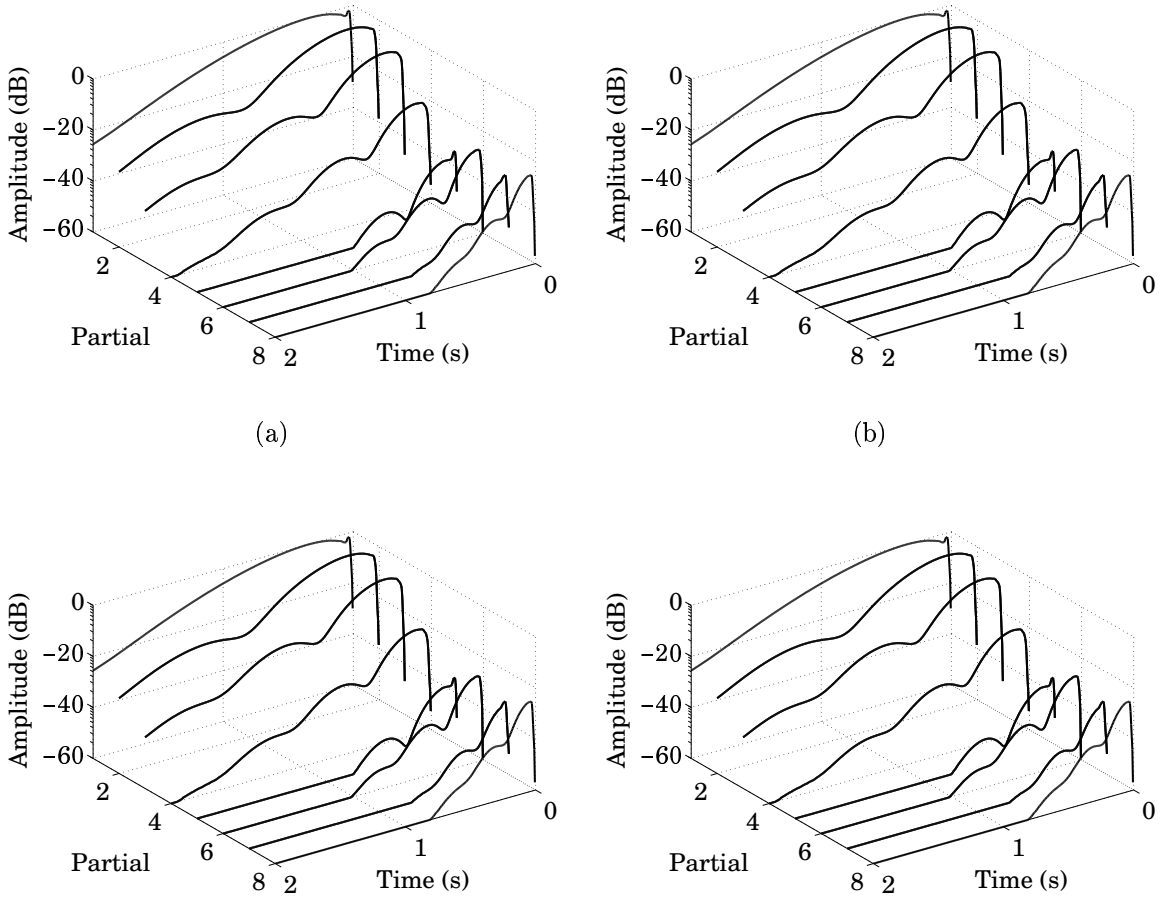


Figure 6.2: Time domain analysis for the synthesized tones according to experiment 1 in Table 6.1. Original excitation is used for the resynthesis. (a) Target tone. (b) Estimated tone, run 1, Error = 0.0465. (c) Estimated tone, run 2, Error = 0.0123. (d) Estimated tone, run 3, Error = 0.0232.



(a)

(b)

(c) Eight first partials for the estimated tone with parameter values according to the run 2 in Table 6.1.

(d) Eight first partials for the estimated tone with parameter values according to the run 3 in Table 6.1.

Figure 6.3: Frequency analysis for the synthesized tones according to experiment 1 Table 6.1. Original excitation is used for the resynthesis. Eight first partials for the tones are presented. (a) Target tone. (b) Estimated tone, run 1, $Error = 0.0465$. (c) Estimated tone, run 2, $Error = 0.0123$. (d) Estimated tone, run 3, $Error = 0.0232$.

The convergence of error for three runs in experiment 1 is shown in Figure 6.4. Error minimum is found in generation 197 in run one, 281 in run two, and 164 in run three. Convergence in runs one and two are quite similar implying the particular minima to be smooth. Three additional runs confirms that a strong local minimum according to first six parameters is found in run one and two. The error value in confirm runs settles in $0.0461 - 0.0468$. Error value in run three is better than earlier and steep drops in Figure 6.4(c) implies better convergence, but the minimum is harder to obtain with additional runs. It can be assumed that the minimum is non smooth.

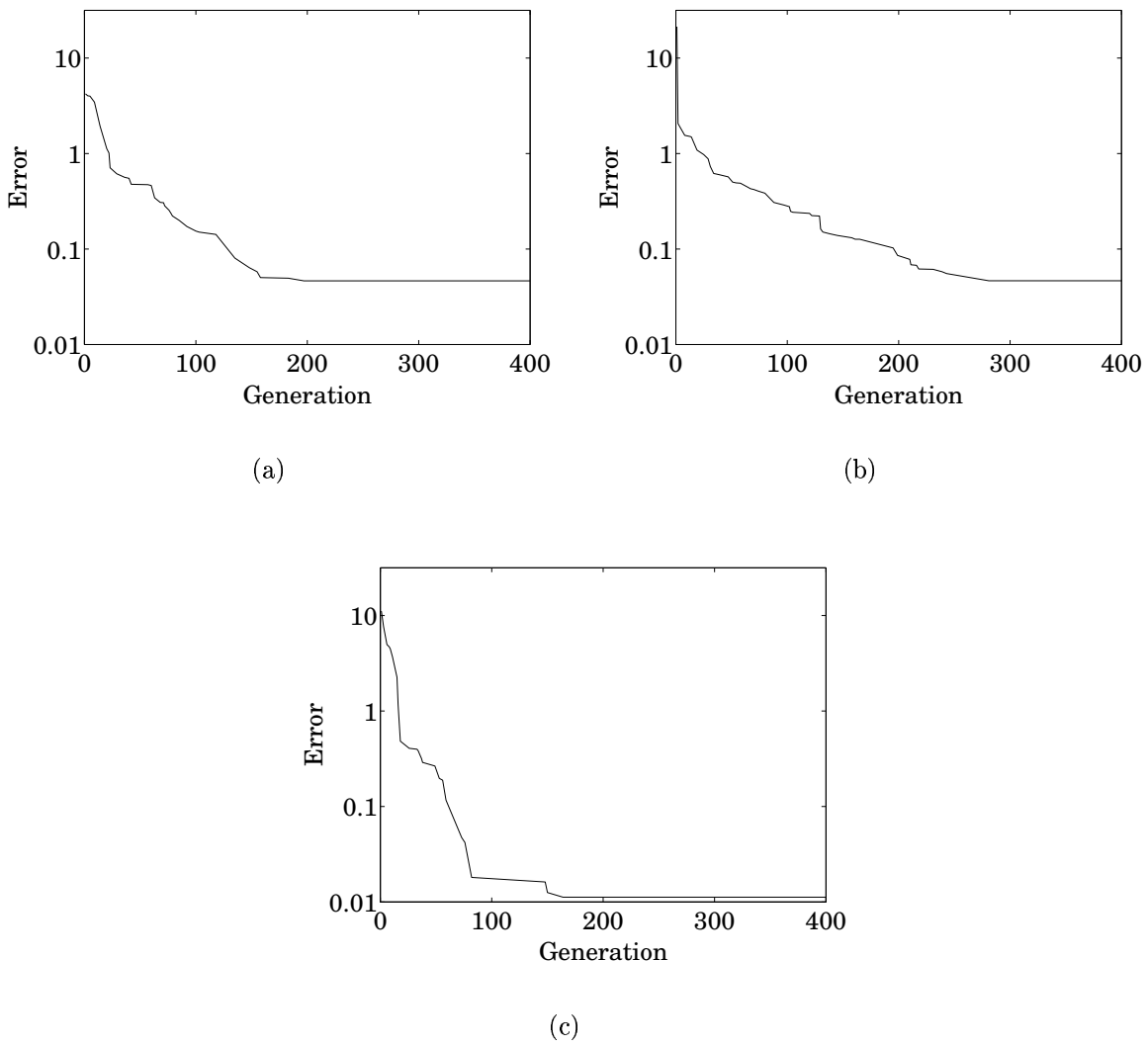


Figure 6.4: Convergence of error in experiment 1. (a) Run 1. (b) Run 2. (c) Run 3.

Although we can estimate parameters which produce tones that are indistinguishable from the target one, the unpunctuality in values indicates that our parameter estimation procedure operates incompletely. Theoretically, the exact parameter values should be found with the method. GAs has been reported to have disadvantages related to the precision of the final solution. Parallelism of GA ensures effective global search, but as shown by many researchers, GA performs poorly in a localized search (Bersini and Renders, 1994; Houck et al., 1996; Mitchell, 1998). The performance of GA can be enhanced with various localized optimizing methods that start from the point that has been determined by GA as optimal. In our case the non-orthogonality of parameters makes the task even more difficult, since various negligible small error minimums can be found over the parameter space. To overcome the non-orthogonality problem we have run the algorithm with constant values of m_p and m_o. The algorithm parameters for the experiment 2 are set as in previous experiment. We have used two different

model parameter settings for the target tone:

1. Parameters are set according to discrete grid, and so zero error is possible with the exact parameter values
2. Parameter values lie between two discrete values just slightly off the midpoint.

Mixing parameters are fixed as $m_p = m_o = 0.5$ and parameter g_c is to be estimated. Results are shown in Table 6.2. The exact parameters are estimated in run 1 and error is obviously zero. The error convergence of run 1 is shown in Figure 6.5(a). Apart from the fact that the parameter values are estimated precisely, the convergence of the algorithm is very fast. Zero error is found already in generation 87. Running the algorithm multiple times ensures the fast convergence. We rerun the algorithm two times and zero error was found in generations 104 and 88. Convergence of run 2 is also fast as can be seen in Figure 6.5(b), where the minimum is found in generation 77. The nearest points in the discrete scales are estimated for the f_0', d_f, g_c, and the decay parameters g_h, g_v, and a_h. Decay parameters are swapped between the polarizations, but as mentioned it is the feature of the model, when $m_p = 1 - m_o$. Optimal error value 0.1971 is more than four times higher than in experiment 1, where the target parameter values match better the discrete grid. As discussed in Section 5.1.4 parameters d_f and g are much more sensitive than parameter a. This might be the reason for the slight deviation in frequency dependant gain a_h. Dominant error due to differences in d_f and g is reduced by adjusting the unsensitive parameter a_h. If a_h is chanced next to the target value, the agreement of parameters is best possible, but the resulting error value 0.3765 is twice as high than in the estimated case. This supports the conclusion above.

parameter	original parameter (run1)	estimated parameter (run1)	original parameter (run2)	estimated parameter (run2)
f_0'	330.5	330.5	330.599	330.6139
d_f	0.8987	0.8987	0.962	1.0272
g_h	0.9873	0.9873	0.9865	0.9916
a_h	-0.2905	-0.2905	-0.3006	-0.1936
g_v	0.9907	0.9907	0.9911	0.9859
a_v	-0.1936	-0.1936	-0.1873	-0.2537
g_c	0.1013	0.1013	0.096	0.1013
error	–	0	–	0.1971

Table 6.2: Experiment 2. Original and estimated parameters when a synthesized tone with known parameter values are used as a target tone. The original and the extracted excitation is used for the resynthesis.

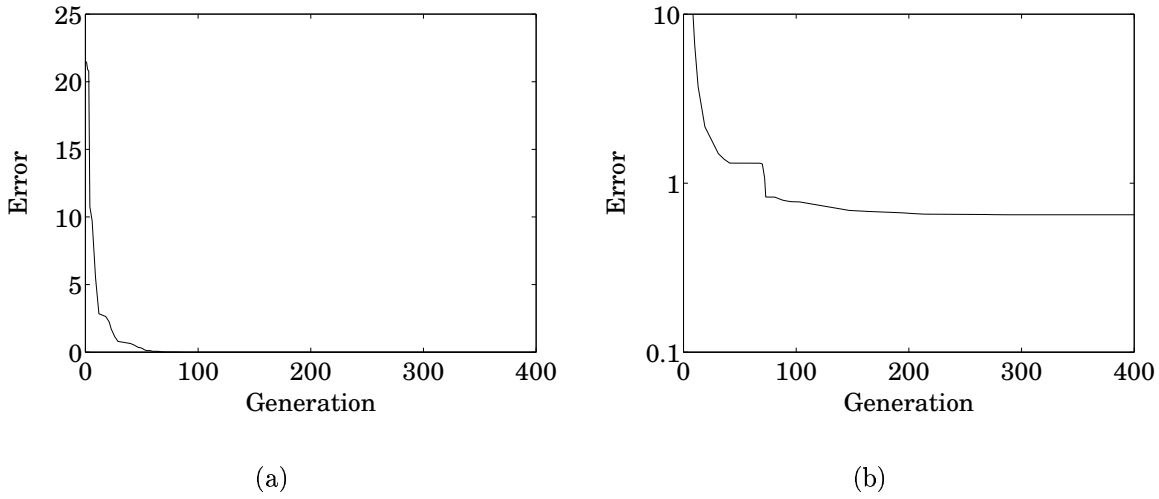
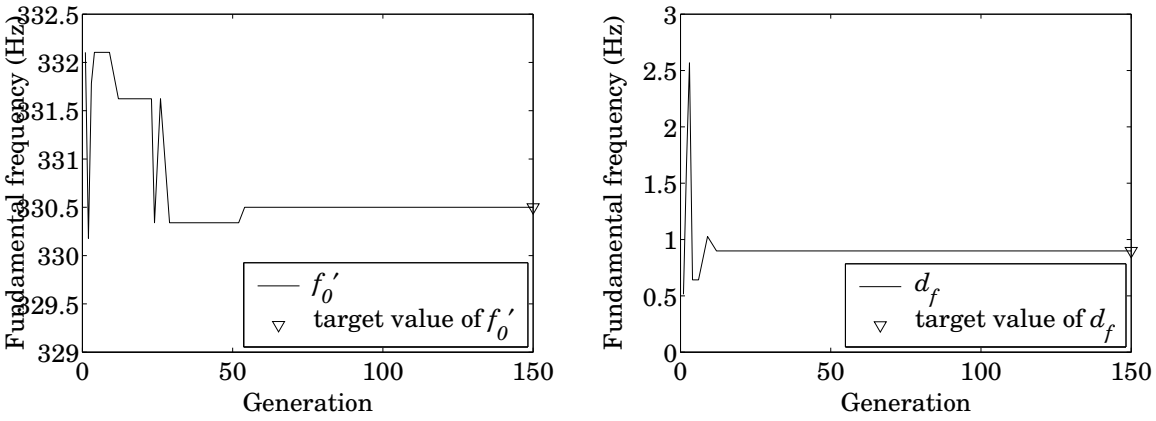


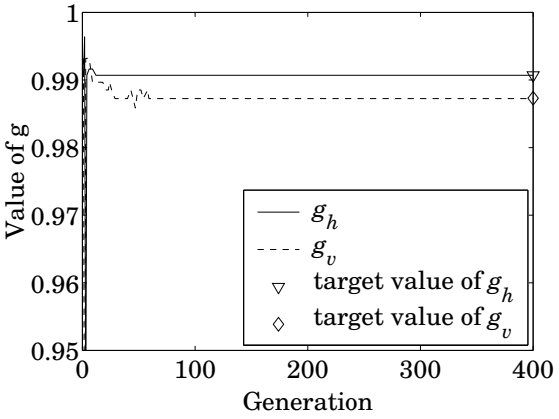
Figure 6.5: Convergence of error in experiment 2. (a) Run 1. (b) Run 2.

The convergence of the parameters for second experiment's run number one is shown in Figure 6.6. The most dominant parameters are difference d_f and overall decay parameters g_h and g_v, which converge first while other parameters keeps vibrating and settle in the later stage of the algorithm.

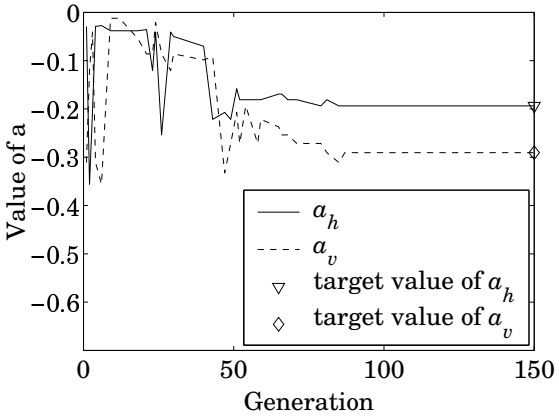


(a)

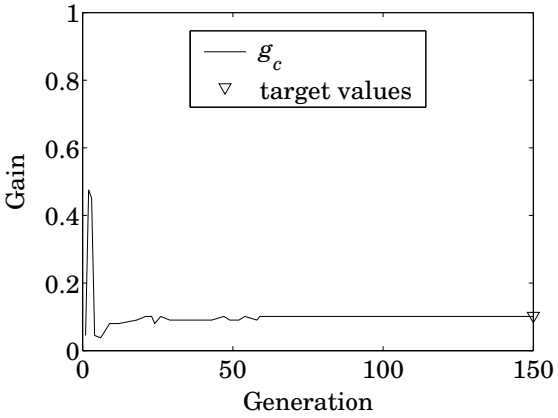
(b)



(c)



(d)



(e)

Figure 6.6: Convergence of the parameters for the first 150 generations of experiment 2 in Table 6.2. An original excitation is used for the resynthesis. (a) f_0'. (b) d_f. (c) g_h and g_v. (d) a_h and a_v. (e) g_c.

6.2 Synthesized Target Tone with an Extracted Excitation

In the third experiment an excitation that is extracted from a target tone is used for the resynthesis. The parameters for the algorithm is set as follows:

> Population size $S_p = 60$, number of generations = 500, probability of selecting the best individual $q = 0.01$, degree of non-uniformity $b = 3$, retries $w = 3$, number of crossovers = 18, Number of mutations = 18.

The original and the estimated parameters for three runs are shown in Table 6.3.

parameter	original parameter	estimated parameter (run1)	estimated parameter (run2)	estimated parameter (run3)
f_0'	330.5409	331.00085	330.51935	331.00085
d_f	0.8986	0.8986	0.8986	0.8986
g_h	0.9873	0.9907	0.9907	0.9907
a_h	-0.2905	-0.2071	-0.1809	-0.1936
g_v	0.9907	0.9873	0.9873	0.9873
a_v	-0.1936	-0.1290	-0.0920	-0.1290
m_p	0.5	1.0000	0.6971	0.9324
m_o	0.5	0.8715	0.8135	0.6511
g_c	0.1013	0.2450	0.1800	0.0703
error	–	0.4131	0.4657	0.4283

Table 6.3: Original and estimated parameters when a synthesized tone with known parameter values are used as a target tone. An extracted excitation is used for the resynthesis.

Similar behavior is noticed when an extracted excitation is used. The difference is estimated precisely and only a small variation in the mean frequency can be noticed. If the decay parameters are swapped we ge exact match for the g_h and g_v. Parameters m_p, m_o and g_c drifts as in experiment one. Interestingly, the $m_p = 1$ in the first run and close to one in third run, which means that the straight path to vertical polarization is totally or nearly closed. The model is, in a manner of speaking, rearranged such a way that the individual string models are in series as opposed to the original construction where the polarization are arranged in parallel.

We can again look at the envelopes and partial envelopes of the tones in Figure 6.7 and 6.8. Envelopes are almost identical. Only slight inaudible dissimilarity in the beating amplitude in partials can be noticed.

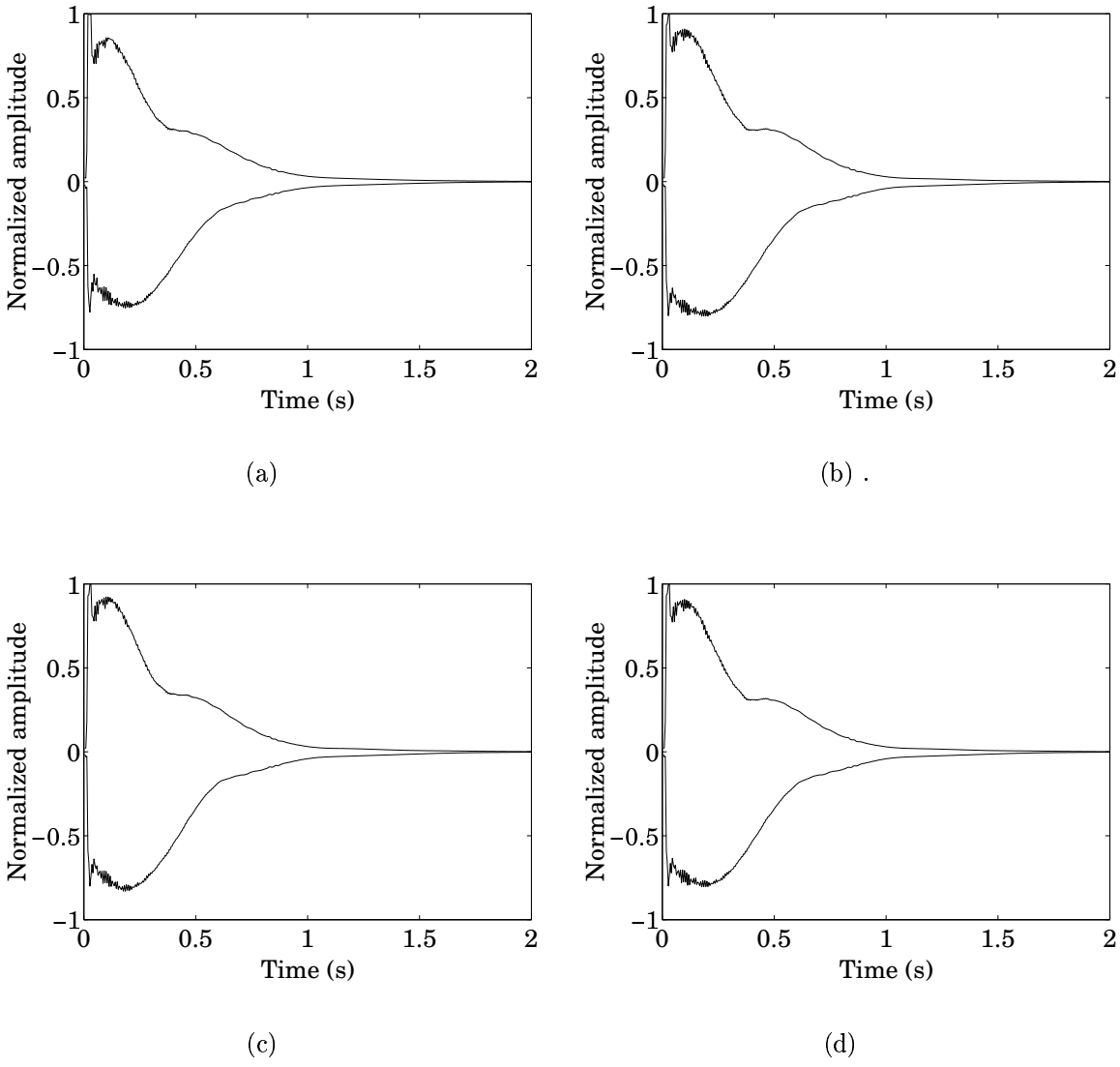


Figure 6.7: Time domain analysis for the synthesized tones according to experiment 3 in Table 6.3. An extracted excitation is used for the resynthesis. (a) Target tone. (b) Estimated tone, run 1, $Error = 0.4131$. (c) Estimated tone, run 2, $Error = 0.4657$. (d) Estimated tone, run 3, $Error = 0.4283$.

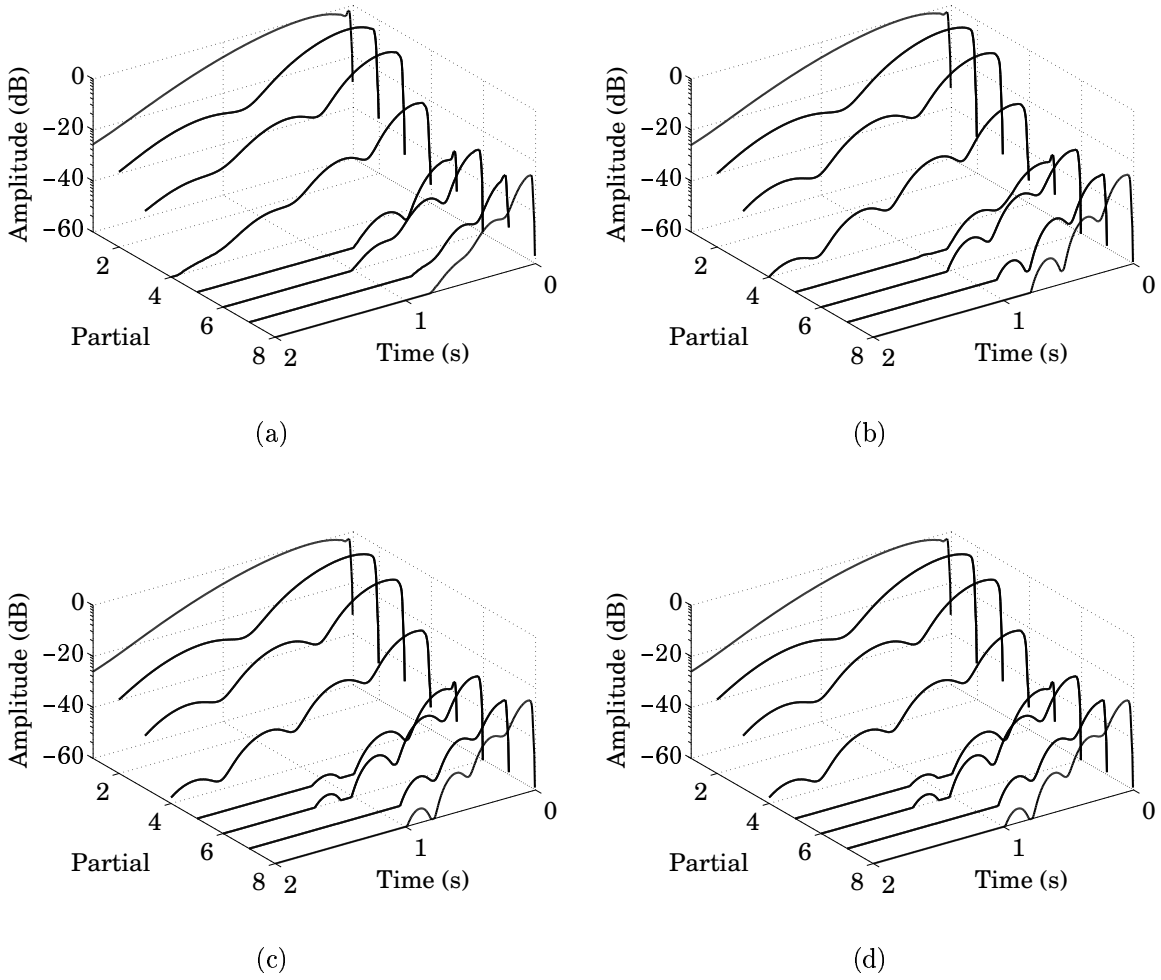


Figure 6.8: Frequency analysis for the synthesized tones according to experiment 1 Table 6.1. An extracted excitation is used for the resynthesis. Eight first partials for the tones are presented. (a) Target tone. (b) Estimated tone, run 1, $Error = 0.4131$. (c) Estimated tone, run 2, $Error = 0.4657$. (d) Estimated tone, run 3, $Error = 0.4283$.

Unlike in the previous experiments the exact parameter values are not so relevant since different excitation signals are used for the target and estimated tones. The initial energy of each partial differs subtly resulting to amplitude envelopes of partials. When used identical parameters and extracted excitation gives 1.6889 error that is four times higher than in experiment three, implying that the error minimum is not found anymore with the target parameters values. Eight first partials for two tones using similar parameter settings and different excitation signals is shown in Figure 6.9. Better agreement in partial envelopes can be obtained with parameters estimated in experiment three. Mixing gains, coupling coefficient and parameters a_h and a_v are adjusted differently to target values to decrease the error. Fixing $m_p = m_o = 0.5$ as in experiment two does not improve the convergence nor the error value. Three additional algorithm runs with mixing parameter fixed gave error values between 0.6779 and

0.6501, implying that the power of the model is slightly lost when mixing parameters are constant. Better results are obtained if input mixing coefficient m_p is constant and m_o is estimated. Fixing $m_p = 1$ gives error 0.4130 in generation 345 and $m_p = 0.5$ gives error 0.4275 in generation 322. Similar results are obtained with multiple runs, indicating better stability of the algorithm and higher orthogonality of parameters. Another question is that which would be the best value for m_p and if the power of expression of the model is limited critically with the particular value in a real case.

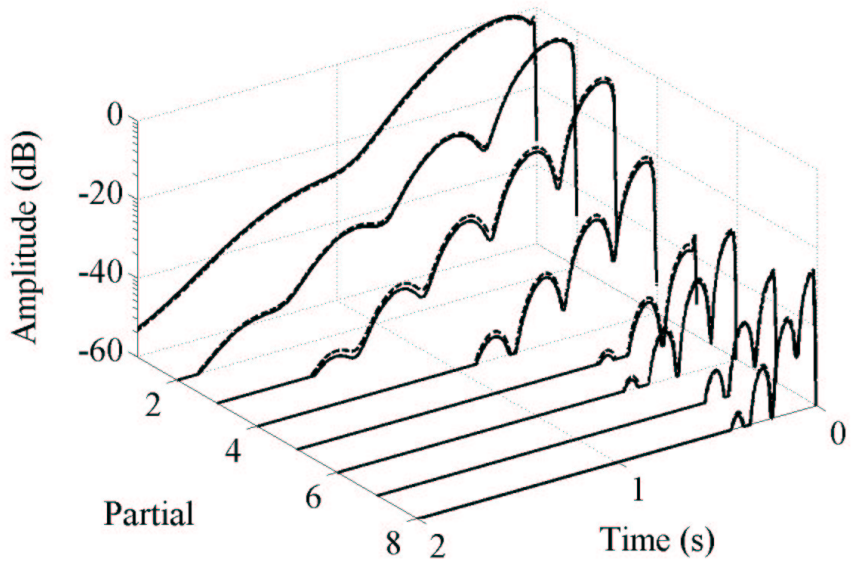


Figure 6.9: Eight first partials for two tones using similar parameter settings and different excitation signals. Solid line shows the partials when the original excitation is used. Dashed line shows the partials when the excitation signal is extracted from the target sound.

The convergence of the error of the third experiment is shown in Figure 6.10 and the convergence of the parameters run for the run number one is shown in Figure 6.11.

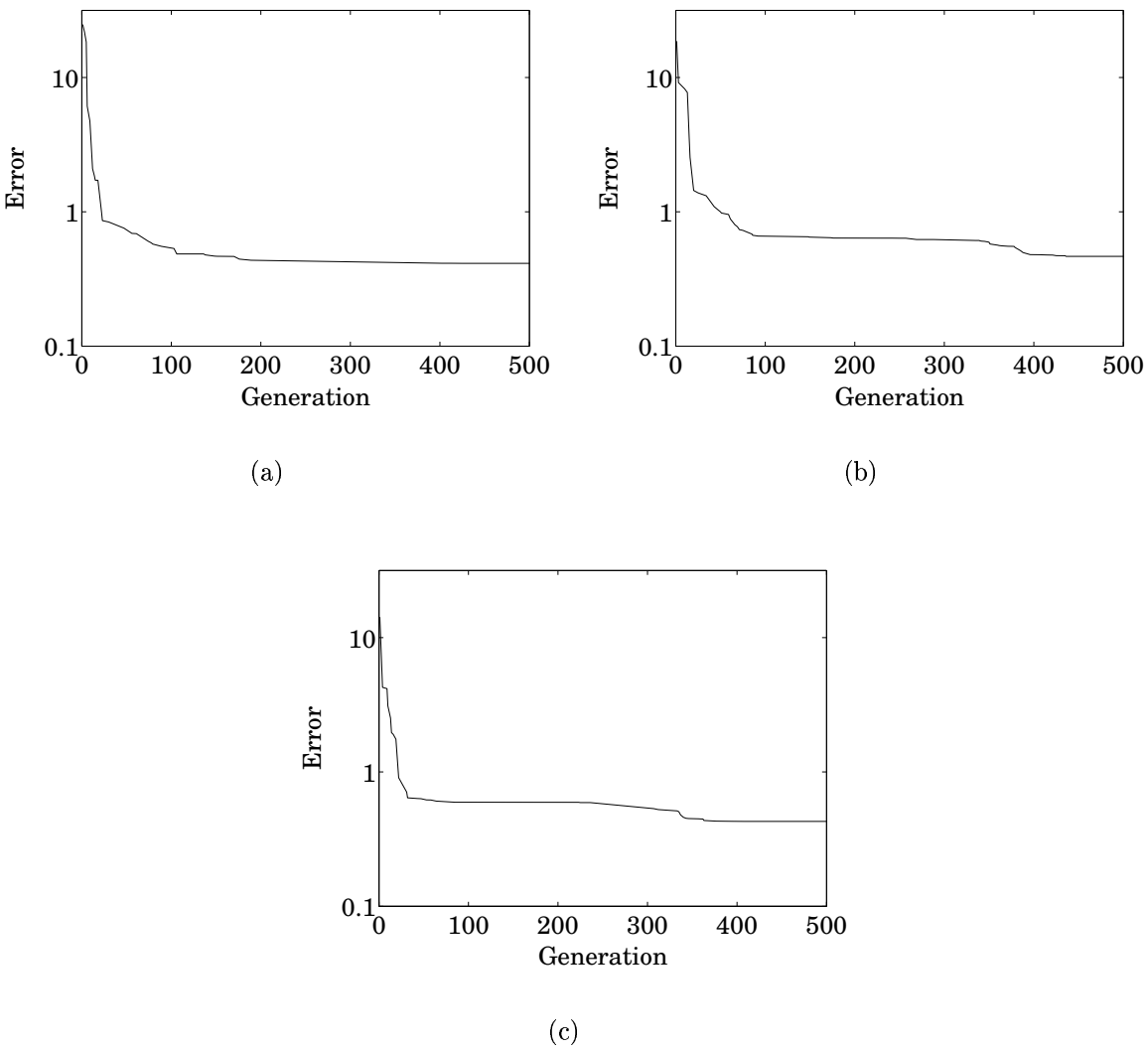
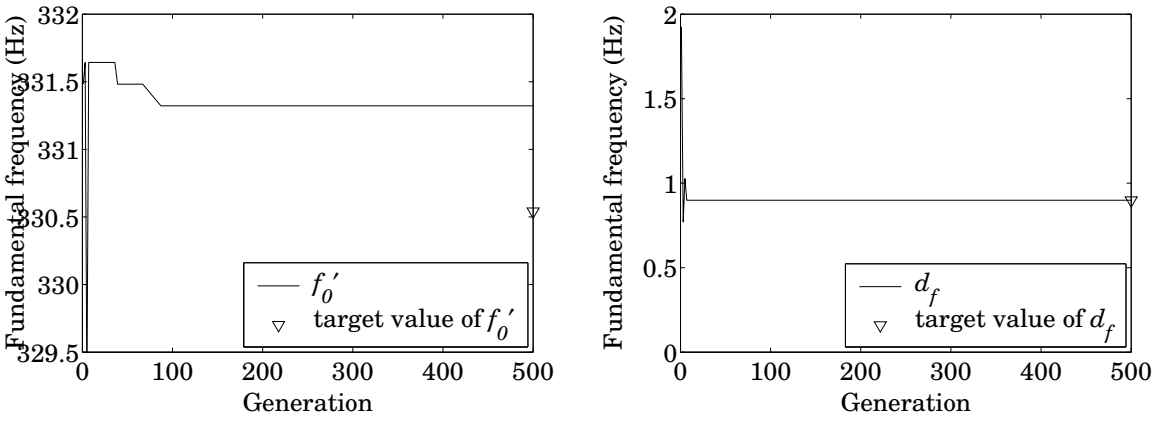


Figure 6.10: Convergence of error in experiment 3. (a) Run 1. (b) Run 2. (c) Run 3.

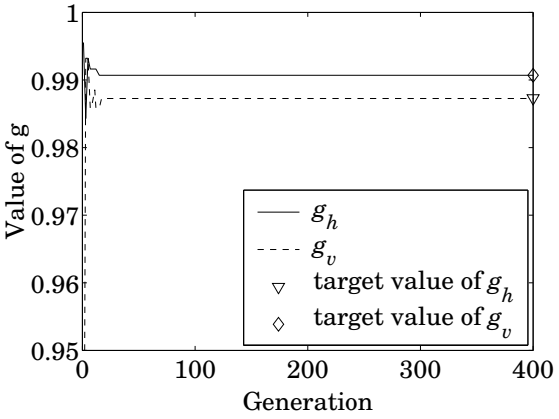
6.3 Recorded Target Tone

Our estimation method is designed to use with real recorded tones. We ha test our system with four tones with fundamental frequencies $\approx$ 330 Hz, 196.5 Hz, 145 Hz, and 82.5 Hz. Time and frequency analysis for such cases are shown in Figures 6.12, 6.13, 6.14,and 6.15. When $f0 = 330$ the envelope and the partials for a recorded tone are very similar with those that are analyzed from a tone that uses estimated parameter values. When looking at the partials of three other cases we can say that the time domain envelopes are very much alike, so the original and synthesized tones are decayed similarly. In Figure 6.15 we can see that the original tone has strong beating in its seventh partial. In the synthesized tone the initial amplitudes and the overall decay characteristics of the partials are similar. Slight beating can be noticed

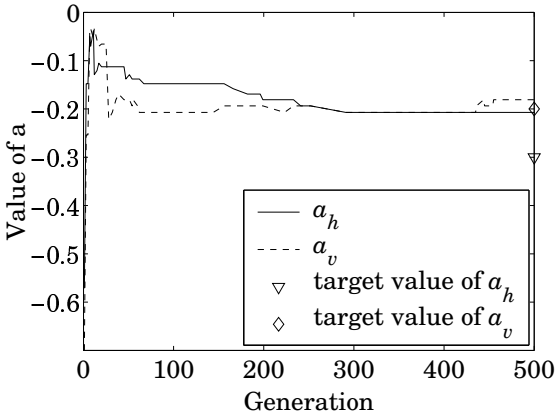


(a)

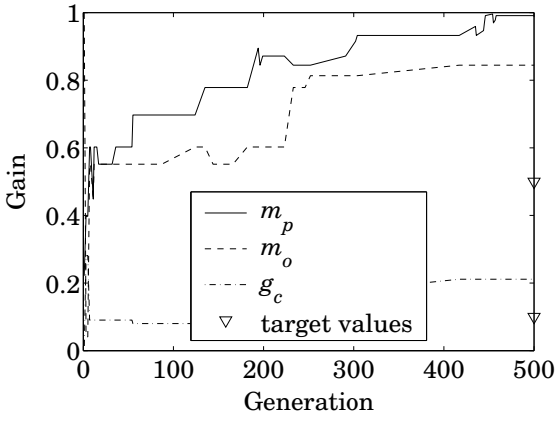
(b)



(c)



(d)



(e)

Figure 6.11: Convergence of the parameters of the experiment 3 in Table 6.3 500 generations are produced and the synthesized excitation is used for the resynthesis. (a) f_0'. (b) d_f. (c) g_h and g_v. (d) a_h and a_v. (e) mp, m_o and g_c.

in all 8 partials which results in the same kind of two stage decay in the time domain envelope.

Appraisal of the perceptual quality of synthesized tones is left as a future project but our informal listening indicates that the quality is comparable or better than with our previous methods and it does not require any hand tuning after the estimation procedure. Sound clips demonstrating these experiments are available in the World Wide Web at `http://www.acoustics.hut.fi/publications/papers/jasp-ga/`.

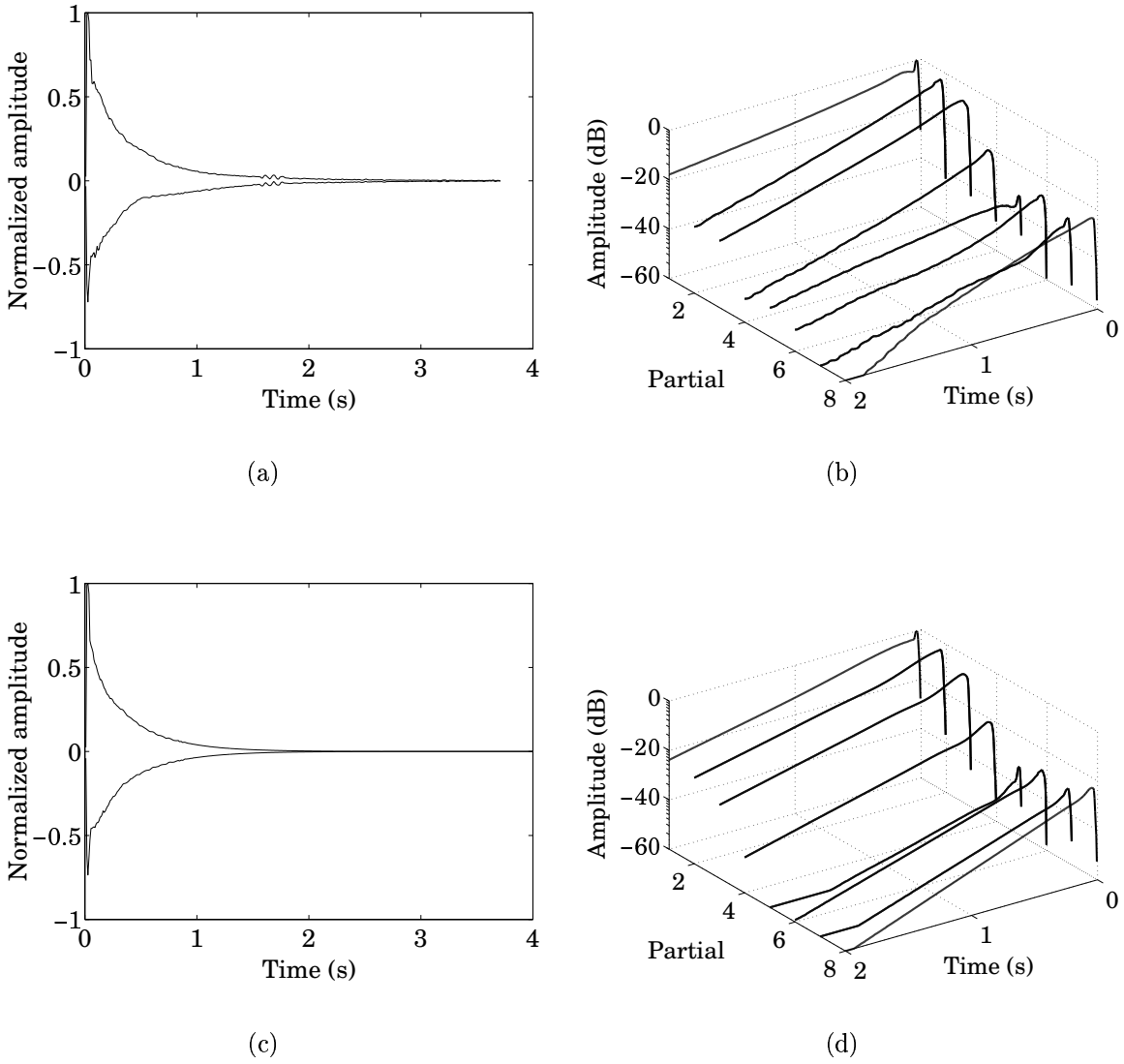


Figure 6.12: Time and frequency analysis for a recorded tone and for a synthesized tone that uses estimated parameter values. Extracted excitation is used for the resynthesis. Estimated parameter values are $f_0' = 331.1044$, $d_f = 1.1558$, $g_h = 0.9762$, $a_h = -0.4991$, $g_v = 0.9925$, $a_v = -0.0751$, $m_p = 0.1865$, $m_o = 0.7397$, and $g_c = 0.1250$. (a) Time domain envelope of a recorded tone. (b) 8 first partials of a recorded tone. (c) Envelope of an estimated tone. (d) 8 first partials of an estimated tone.

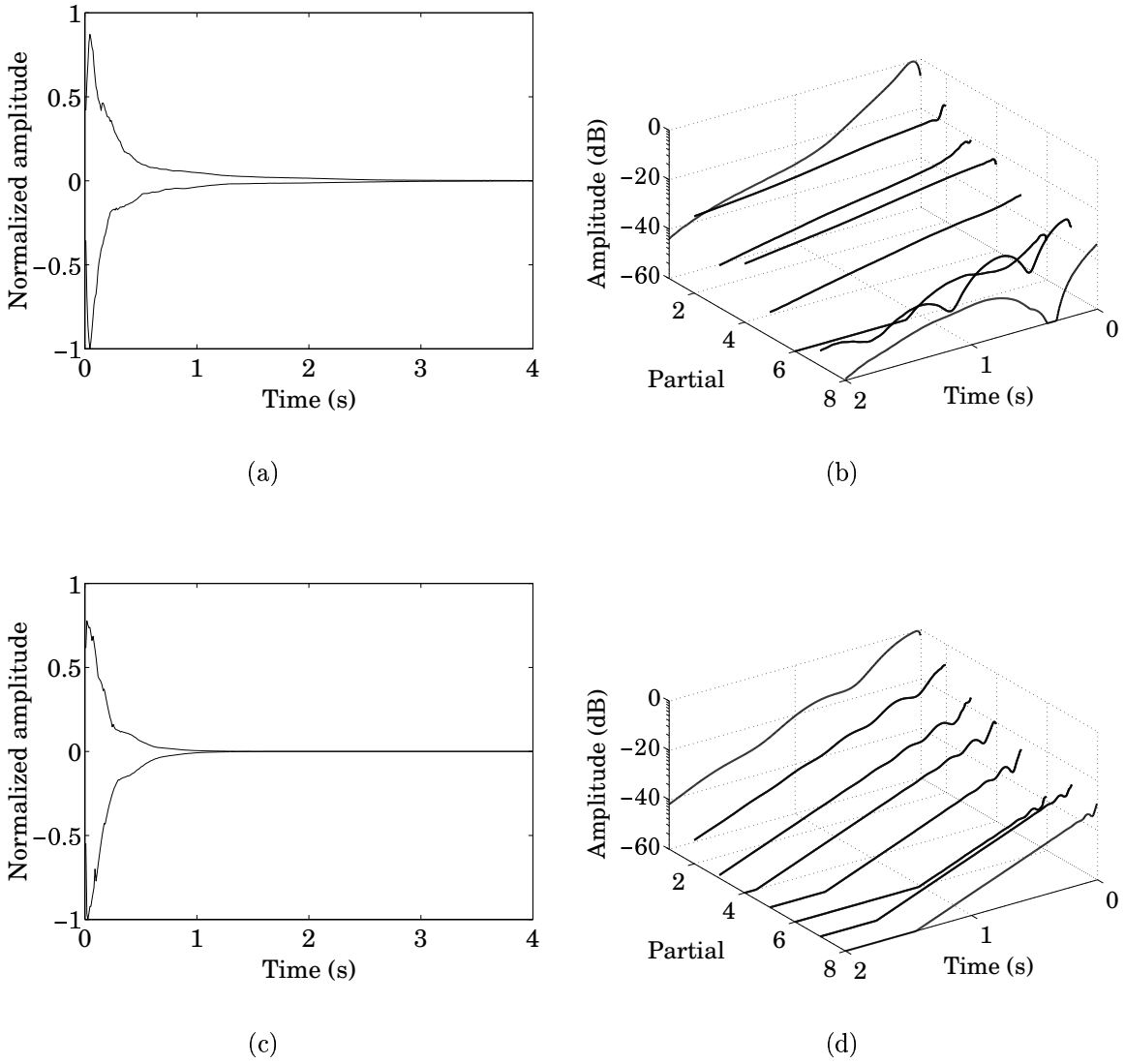


Figure 6.13: Time and frequency analysis for a recorded tone and for a synthesized tone that uses estimated parameter values. Extracted excitation is used for the resynthesis. Estimated parameter values are $f_0' = 197.86765$, $d_f = 2.6985$, $g_h = 0.9763$, $a_h = -0.0054$, $g_v = 0.9762$, $a_v = -0.7000$, $m_p = 0.7397$, $m_o = 0.4483$, and $g_c = 0.0378$. (a) Time domain envelope of a recorded tone. (b) 8 first partials of a recorded tone. (c) Envelope of an estimated tone. (d) 8 first partials of an estimated tone.

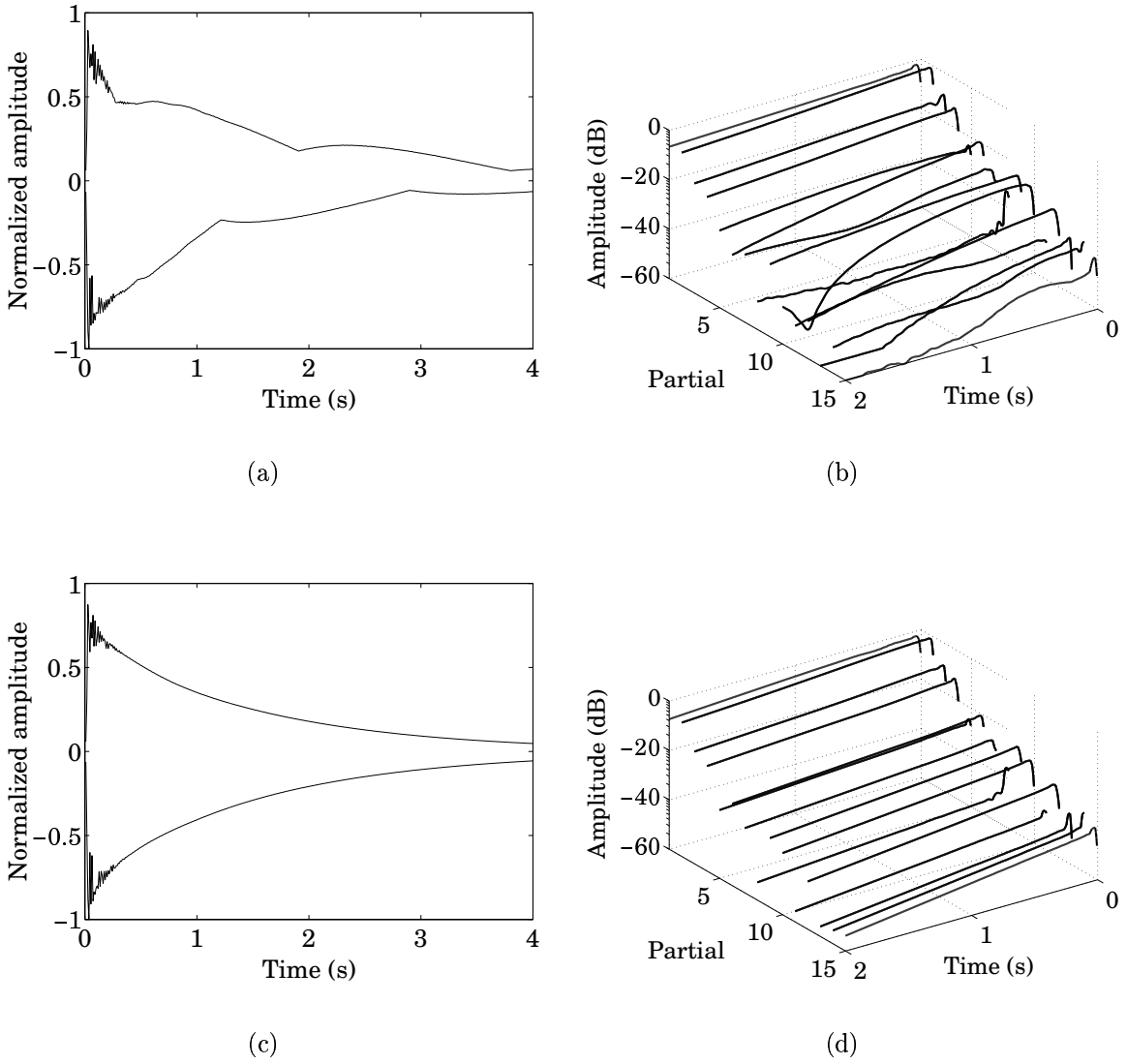


Figure 6.14: Time and frequency analysis for a recorded tone and for a synthesized tone that uses estimated parameter values. Extracted excitation is used for the resynthesis. Estimated parameter values are $f_0' = 146.16$, $d_f = 0.3916$, $g_h = 0.9957$, $a_h = -0.1127$, $g_v = 0.9831$, $a_v = -0.7000$, $m_p = 0.3489$, $m_o = 0.94693$, and $g_c = 0.0903$. (a) Time domain envelope of a recorded tone. (b) 8 first partials of a recorded tone. (c) Envelope of an estimated tone. (d) 8 first partials of an estimated tone.

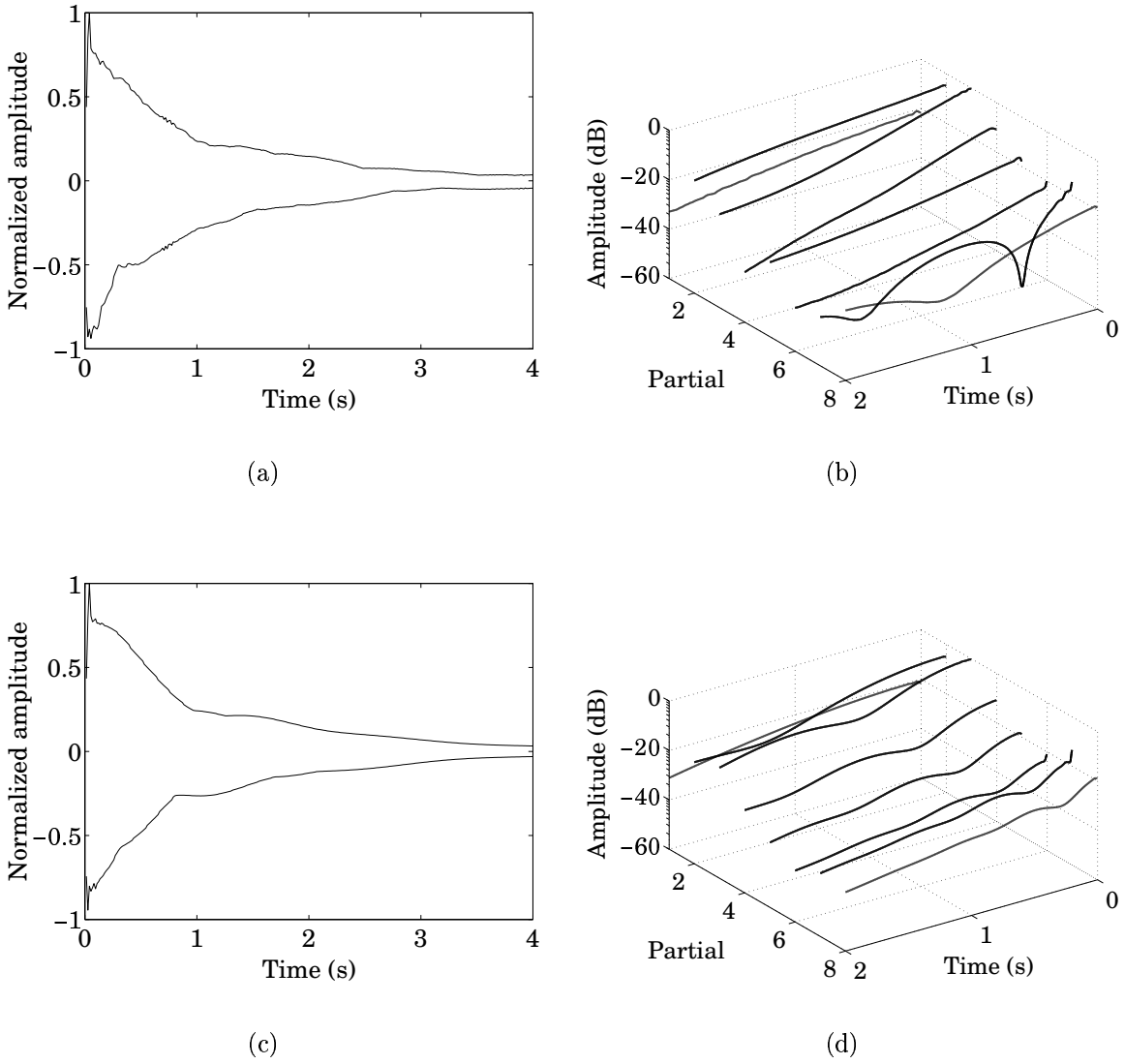


Figure 6.15: Time and frequency analysis for a recorded tone and for a synthesized tone that uses estimated parameter values. Extracted excitation is used for the resynthesis. Estimated parameter values are $f_0' = 81.9445$, $d_f = 0.2423$, $g_h = 0.9915$, $a_h = -0.2371$, $g_v = 0.9883$, $a_v = -0.7000$, $m_p = 0.8715$, $m_o = 0.2215$, and $g_c = 0.0050$. (a) Time domain envelope of a recorded tone. (b) 8 first partials of a recorded tone. (c) Envelope of an estimated tone. (d) 8 first partials of an estimated tone.

Chapter 7

Conclusions and Future Work

In this thesis, a procedure for estimating the parameters of a plucked string synthesis algorithm is studied and designed. The procedure is based on a genetic algorithm. Different genetic operators and methods are first surveyed and a suitable algorithm for our purposes is designed.

Genetic algorithm is a universal optimizing tool, which is well suited to our problem. The most important question is the fitness calculation. How to rank sounds? Employing the knowledge of properties of the human auditory system, a psychoacoustic model for fitness calculation is designed. The frequency-dependent sensitivity and frequency masking of human hearing are taken into account in the model. In addition, the parameter space is discretized in a perceptually reasonable manner. The non-uniform discrete sampling grid for all parameters is designed based on former research results, experiments on parameter sensitivity, and informal listening.

The estimation method is designed for use in natural-sounding synthesis of various string instruments. All these instruments have their own sound characteristics that have to be included in synthesized tones by defining correct parameter values. In previous methods some parameters have had to be fine-tuned manually by an expert user. This has been a clear disadvantage. The objective of this work was to create a fully automated method for parameter estimation and to improve the quality of previous methods. These two goals were reached, but also more supplementary studies have to be carried out in the future.

The system has been tested with both synthetic and real recordings. When using synthetic tones as target tones we are able to evaluate the parameter estimation procedure since the target values of parameters are known. This parameter set exactly reproduces the target tone and therefore gives zero error when analyzed with the error calculation function. In practice the zero error cannot be reached if some of the target values are

not adjusted with the discrete grid. The method has been tested with two choices of excitation signals for the plucked string synthesis model. A parameter set with zero error (if target values are adjusted with the grid) or negligible error can be found in both cases. Original and synthesized tones are indistinguishable. The estimation method is designed to use with real recording. Since there are no known correct parameter values, the quality of the estimation method is evaluated by comparing the target and synthesized tones. This is carried out in time and frequency domain by examining the time domain envelopes and partial envelopes of the tones. The estimation procedure works great with simple tones that is when all the partials behave similarly. The procedure also works with complex tones, where individual partials may have dissimilar behavior. Although the result is not exactly identical with the target tone the synthesized tones sound realistic. The implementation of the parameter estimation method and the results of our studies have been published in the references (Riionheimo and Välimäki, 2002) and (Riionheimo and Välimäki, 2003).

Perceptual quality of resynthesis is difficult to measure. An accurate way to measure the sound quality would be to carry out listening tests with trained participants. Appraisal of synthetic tones that use parameter values from the proposed GA-based method is left as a future project. More perceptual studies has to be carried out and better auditory models for the fitness calculation have to be designed.

Bibliography

Allen, J. B. and Rabiner, L. R. (1977). A unified approach to short-time fourier analysis and synthesis. In *Proceedings of the IEEE*, pages 1558–1564.

Bank, B. (2000). Physics-based sound synthesis of the piano. *Helsinki University of Technology, Laboratory of Acoustics and Audio Signal Processing.* Report no. 54, Espoo, Finland, 2000.

Bank, B., Välimäki, V., Sujbert, L., and Karjalainen, M. (2000). Efficient physics-based sound synthesis of the piano using DSP methods. In *Proceedings of the European Signal Processing Conference (EUSIPCO 2000)*, pages 2225–2228, Tampere, Finland.

Bersini, H. and Renders, B. (1994). Hybridizing genetic algorithms with hill-climbing methods for global optimization: Two possible ways. In *Proceedings of the IEEE International Symposium Evolutionary Computation*, pages 312–317, Orlando, Florida, USA.

Biles, J. (1994). Genjam: A genetic algorithm for generating jazz solos. In *Proceedings of the 1994 ICMC*, pages 131–137.

Blickle, T. and Thiele, L. (1995). A comparison of selection schemes used in genetic algorithms. *Computer engineering and communication networks lab (TIK), Swiss federal institute of Technology (ETH).* Technical Report.

De Jong, K. A. (1975). An analysis of the behavior of a class of genetic adaptice systems (Doctoral dissetation, University of Michigan). *Dissertation Abstract International*, 36. University Microfilms No 76-9381.

Drioli, C. and Rocchesso, D. (1998). Learning pseudo-physical models for sound synthesis and transformation. In *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, pages 1085 – 1090, San Diego, CA, USA.

Erkut, C., Laurson, M., Kuuskankare, M., and Välimäki, V. (2001). Model-based synthesis of the ud and the Renaissance lute. In *Proceedings of the International Computer Music Conference (ICMC 2001)*, Havana, Cuba.

Erkut, C. and Välimäki, V. (2000). Model-based sound synthesis of tanbur, a Turkish long-necked lute. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP'00)*, pages 769–772, Istanbul, Turkey.

Erkut, C., Välimäki, V., Karjalainen, M., and Laurson, M. (2000). Extraction of physical and expressive parameters for model-based sound synthesis of the classical guitar. *Presented at the AES 108th Convention*, preprint 5114. Paris, France.

Eshelman, L., Caruana, R., and Schaffer, J. (1989). Biases in the crossover landscape. In *Proceedings of the Third International conference on genetic algorithms*, pages 10–19, San Mateo, California, USA.

Fletcher, H. and Munson, W. (1933). Loudness, definition, measurement and calculation. *Journal of the Acoustical Society of America*, 6:82–108.

Garcia, G. and Pampin, J. (1999). Data compression of sinusoidal modeling parameters based on psychoacoustic masking. In *Proceedings of the International Computer Music Conference (ICMC 1999)*, pages 40–43, Beijing, China.

Garcia, R. (1998). Automatic generation of sound synthesis techniques. Master's thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA.

Goldberg, D., Dep, K., and Korb, B. (1991). Do not worry, be messy. In *Proceedings of the Fourth International Conference on Genetic Algorithms*, pages 24–30, San Mateo, California, USA.

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, Reading, Massachusetts, USA, 1989.

Grefenstette, J. J. and Baker, J. E. (1989). How genetic algorithms work: A critical look at implicit parallelism. In *Proceedings of the Third International Conference on Genetic Algorithms*, pages 20–27, San Mateo, California, USA.

Hermus, K., Verhelst, W., and Wambacq, P. (2002). Psycho-acoustic modeling of audio with exponentially damped sinusoids. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2002)*, pages 1821–1824, Orlando, Florida, USA.

Holland, J. (1975). *Adaptation in Natural and Artificial Systems*. The University of Michigan Press, Ann Arbor, USA, 1975.

Horner, A., Beauchamp, J., and Haken, L. (1993). Machine tonques 16: Genetic algorithms and their application to FM matching synthesis. *Computer Music Journal*, 17:17–29.

Houck, C., Joines, J., and Kay, M. (1996). Comparison of genetic algorithms, random restart, and two-opt switching for solving large location-allocation problems. *Computers and Operations Research*, 23:587–596.

ISO/IEC 11172-3 (1993). Information technology - coding of moving pictures and associated audio for digital storage media at up to about 1.5 mbit/s - part 3: Audio. *ISO/IEC*.

Jaffe, D. and Smith, J. O. (1983). Extensions of the Karplus-Strong plucked string algorithm. *Computer Music Journal*, 7:43–55.

Järveläinen, H. and Tolonen, T. (2001). Perceptual tolerances for decay parameters in plucked string synthesis. *Journal of the Audio Engineering Society*, 49:1049–1059.

Johnson, C. (1999). Exploring the sound-space of synthesis algorithms using interactive genetic algorithms. In *Proceedings of the AISB Workshop on Articial Intelligence and Musical Creativity*, pages 20–27, Edinburgh, Scotland.

Johnston, J. D. (1988a). Estimation of perceptual entropy using noise masking criteria. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'88)*, pages 2524–2527, New York, NY, USA.

Johnston, J. D. (1988b). Transform coding of audio signals using perceptual noise criteria. *IEEE Journal on Selected Areas in Communications*, 6:314–323.

Joines, J. and Houck, C. (1994). On the use of non-stationary penalty functions to solve constrained optimization problems with genetic algorithms. In *Proceedings of the IEEE International Symposium Evolutionary Computation*, pages 579–584, Orlando, Florida, USA.

Karjalainen, M. (1999). Kommunikaatioakustiikka. *Helsinki University of Technology, Laboratory of Acoustics and Audio Signal Processing*. Report no. 51, Espoo, Finland, 1999.

Karjalainen, M., Välimäki, V., and Jánosy, Z. (1993). Towards high-quality sound synthesis of the guitar and string instruments. In *Proceedings of the International Computer Music Conference (ICMC 1993)*, pages 56–63, Tokyo, Japan.

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

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

Lagrange, M. and Marchand, S. (2001). Real-time additive synthesis of sound by taking advantage of phychoacoustics. In *Proceedings of the COST-G6 Conference on Digital Audio Effects (DAFx01)*, Limerick, Ireland.

Laurson, M., Erkut, C., Välimäki, V., and Kuuskankare, M. (2001). Methods for modeling realistic playing in acoustic guitar synthesis. *Computer Music Journal*, 25:38–49.

Liang, S. and Su, A. W. Y. (2000). Recurrent neural-network-based physical model for the chin and other plucked-string instruments. *Journal of the Audio Engineering Society*, 48:1045–1059.

Mattila, V. V. and Zacharov, N. (2001). Generalized listener selection (GLS) procedure. *Presented at the AES 110th Convention*, preprint 5405. Amsterdam, The Netherlands.

Michalewicz, Z. (1992). *Genetic Algorithms + Data Structures = Evolution Programs.* AI Series. Springer-Verlag, New York, USA, 1992.

Mitchell, M. (1998). *An Introduction to Genetic Algorithms.* A Bradford Book, The MIT Press, Cambridge, Massachusetts, USA, 1998.

Mourjopoulos, J. and Tsoukalas, D. (1992). Neural network mapping to subjective spectra of music sounds. *Journal of the Audio Engineering Society*, 40.

Mühlenbein, H. and Schlierkamp-Voosen, D. (1993). Predictive models for the breeder genetic algorithm. *Evolutionary Computation*, 1:25–49.

Nackaerts, A., Moor, B. D., and Lauwereins, R. (2001). Parameter estimation for dual-polarization plucked string models. In *Proceedings of the International Computer Music Conference (ICMC 2001)*, Havana, Cuba.

Painter, T. and Spanias, A. (2000). Perceptual coding of digital audio. In *Proceedings of the IEEE*, pages 451–515.

Papadopoulos, G. (1998). A genetic algorithm for the generation of jazz melodies. In *Proceedings of STeP'98: 8th Finnish Conference on Artificial Intelligence*, Jyväskylä, Finland.

Press, W. H. (1992). *Numerical Recipes of C: The Art of Scientific Computing.* Cambridge University Press, Cambridge, New york, USA, 1992.

Qi, D. and Sun, R. (2001). GA-based multi-agent reinforcement learning for playing backgammon. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)*, pages 20–27, San Francisco, California, USA.

Riionheimo, J. and Välimäki, V. (2002). Parameter estimation of a plucked string synthesis model with genetic algorithm. In *Proceedings of the International Computer Music Conference (ICMC 2002)*, pages 283–286, Göteborg, Sweden.

Riionheimo, J. and Välimäki, V. (2003). Parameter estimation of a plucked string synthesis model using a genetic algorithm with perceptual fitness calculation. *EURASIP Journal on Applied Signal Processing*, 3:791–805. Special issue on Genetic and Evolutionary Computation for Signal Processing and Image Analysis.

Roads, C. (1996). *The Computer Music Tutorial*. The MIT Press, Cambridge, Massachusetts, USA, 1996.

Schaffer, J., Caruana, R., Eshelman, L., and Das, R. (1989). A study of control parameters affecting online performance of genetic algoritms for function optimization. In *Proceedings of the Third International conference on genetic algorithms*, pages 51–60, San Mateo, California, USA.

Schroeder, M., Atal, B. S., and Hall, J. (1979). Optimizing digital speech coders by exploiting masking properties of the human ear. *Journal of the Acoustical Society of America*, 66.

Smith, J. O. (1992). Physical modeling using digital waveguides. *Computer Music Journal*, 16:74–91.

Smith, J. O. (1993). Efficient synthesis of stringed musical instruments. In *Proceedings of the International Computer Music Conference (ICMC 1993)*, pages 64–71, Tokyo, Japan.

Tolonen, T. and Välimäki, V. (1997). Automated parameter extraction for plucked string synthesis. In *Proceedings of the International Symposium on Musical Acoustics (ISMA'97)*, pages 245–250, Edinburgh, Scotland.

Välimäki, V., Huopaniemi, J., Karjalainen, M., and Jánosy, Z. (1996). Physical modeling of plucked string instruments with application to real-time sound synthesis. *Journal of the Audio Engineering Society*, 44:331–353.

Välimäki, V., Karjalainen, M., Tolonen, T., and Erkut, C. (1999). Nonlinear modeling and synthesis of the kantele – a traditional Finnish string instrument. In *Proceedings of the International Computer Music Conference (ICMC 1999)*, Beijing, China.

Välimäki, V. and Tolonen, T. (1998). Development and calibration of a guitar synthesis. *Journal of the Audio Engineering Society*, 46:766–788.

Vuori, J. and Välimäki, V. (1993). Parameter estimation of non-linear physical models by simulated evolution – application to the flute model. In *Proceedings of the International Computer Music Conference (ICMC 1993)*, pages 402–404, Tokyo, Japan.

Weinreich, G. (1977). Coupled piano strings. *Journal of the Acoustical Society of America*, 62:1474–1484.

Wier, C., Jesteadt, W., and Green, D. (1977). Frequency discrimination as a function of frequency and sensation level. *Journal of the Acoustical Society of America*, 61(1):178–184.

Wun, C. W. and Horner, A. (2001). Perceptual wavetable matching for synthesis of musical instrument tones. *Journal of the Audio Engineering Society*, 49:250–262.

Wun, C. W., Horner, A., and Ayers, L. (2001). Perceptual wavetable matching for synthesis of musical instrument tones. In *Proceedings of the International Computer Music Conference (ICMC 2001)*, pages 219–226, Havana, Cuba.

Zwicker, E. and Fastl, H. (1990). *Psychoacoustics: Facts and Models*. Springer-Verlag, Berlin, Germany, 1990.

Zwicker, E. and Zwicker, U. T. (1991). Audio engineering and psychoacoustics: Matching signals to the final receiver, the human auditory system. *Journal of the Audio Engineering Society*, 39.