

**BRNO UNIVERSITY OF TECHNOLOGY**  
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION  
DEPARTMENT OF TELECOMMUNICATIONS

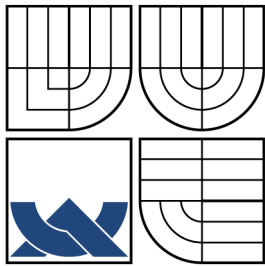
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ  
ÚSTAV TELEKOMUNIKACÍ

**REAL-TIME DIGITAL SIMULATION OF GUITAR  
AMPLIFIERS AS AUDIO EFFECTS**

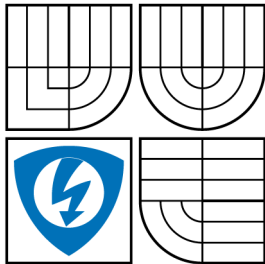
DOCTORAL THESIS  
DIZERTAČNÍ PRÁCE

AUTHOR  
AUTOR PRÁCE

Ing. JAROMÍR MAČÁK



BRNO UNIVERSITY OF TECHNOLOGY  
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ



FACULTY OF ELECTRICAL ENGINEERING AND  
COMMUNICATION  
DEPARTMENT OF TELECOMMUNICATIONS

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH  
TECHNOLOGIÍ  
ÚSTAV TELEKOMUNIKACÍ

## REAL-TIME DIGITAL SIMULATION OF GUITAR AMPLIFIERS AS AUDIO EFFECTS

ČÍSLICOVÁ SIMULACE KYTAROVÝCH ZESILOVAČŮ JAKO ZVUKOVÝCH EFEKTŮ  
V REÁLNÉM ČASE

DOCTORAL THESIS  
DIZERTAČNÍ PRÁCE

AUTHOR  
AUTOR PRÁCE

Ing. JAROMÍR MAČÁK

SUPERVISOR  
VEDOUCÍ PRÁCE

Ing. JIŘÍ SCHIMMEL, Ph.D.

BRNO 2012

## **ABSTRACT**

The work deals with the real-time digital simulation of guitar amplifiers considered as nonlinear analog audio effects. The main aim is to design algorithms which are able to simulate complex systems in real-time. These algorithms are mainly based on the automated DK-method and the approximation of nonlinear functions. Quality of the designed algorithms is evaluated using listening tests.

## **KEYWORDS**

Nonlinear dynamic system, real-time digital signal processing, digital simulation, audio effect, guitar amplifier.

## **ABSTRAKT**

Práce se zabývá číslicovou simulací kytarových zesilovačů, jakož to nelineárních analogových hudebních efektů, v reálném čase. Hlavním cílem práce je návrh algoritmů, které by umožnily simulaci složitých systémů v reálném čase. Tyto algoritmy jsou převážně založeny na automatizované DK-metodě a aproximaci nelineárních funkcí. Kvalita navržených algoritmů je stanovována pomocí poslechových testů.

## **KLÍČOVÁ SLOVA**

Nelineární setrvačné systémy, zpracování číslicových signálů v reálném čase, digitální simulace, hudební efekt, kytarový zesilovač.

MAČÁK, Jaromír *Real-time Digital Simulation of Guitar Amplifiers as Audio Effects*: doctoral thesis. Brno: Brno University of Technology, Faculty of Electrical Engineering and Communication, Department of Telecommunications, 2012. 198 p. Supervised by Ing. Jiří Schimmel, Ph.D.

## DECLARATION

I declare that I have written my doctoral thesis on the theme of “Real-time Digital Simulation of Guitar Amplifiers as Audio Effects” independently, under the guidance of the doctoral thesis supervisor and using the technical literature and other sources of information which are all quoted in the thesis and detailed in the list of literature at the end of the thesis.

As the author of the doctoral thesis I furthermore declare that, as regards the creation of this doctoral thesis, I have not infringed any copyright. In particular, I have not unlawfully encroached on anyone’s personal and/or ownership rights and I am fully aware of the consequences in the case of breaking Regulation § 11 and the following of the Copyright Act No 121/2000 Sb., and of the rights related to intellectual property right and changes in some Acts (Intellectual Property Act) and formulated in later regulations, inclusive of the possible consequences resulting from the provisions of Criminal Act No 40/2009 Sb., Section 2, Head VI, Part 4.

Brno .....

.....

(author’s signature)

## ACKNOWLEDGEMENT

Firstly I would like to thank to my supervisor Jiří Schimmel for this interesting topic and for his valuable advices and remarks which I was given during working on this topic. I also thank to colleagues from my office for inspiring working environment.

Further I would like to thank to Audiffex company for the opportunity to implement algorithms for the simulation of guitar analog effects in their products, thanks to Lubor Příklad, CEO of Audiffex, and to Vladimír Tichý, software developer of Audiffex.

I have to thank to prof. Udo Zölzer and Martin Holters for the opportunity to spend three amazing months at Helmut Schmidt University in Hamburg. This visit inspired me during last stages of my research.

I also have to thank to Oliver Kröning for measuring the tubes from the simulated preamp and to all who took part in the listening tests.

My greatest thanks belong to my family and girlfriend Dana for support and patience during writing this thesis.



Faculty of Electrical Engineering  
and Communication  
Brno University of Technology  
Purkynova 118, CZ-61200 Brno  
Czech Republic  
<http://www.six.feec.vutbr.cz>

## ACKNOWLEDGEMENT

The research was performed in laboratories supported by the SIX project; the registration number CZ.1.05/2.1.00/03.0072, the operational program Research and Development for Innovation.

Brno .....

.....

(author's signature)



EVROPSKÁ UNIE  
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ  
INVESTICE DO VAŠÍ BUDOUCNOSTI



# CONTENTS

<b>List of abbreviations</b>	<b>14</b>
<b>List of symbols and math operations</b>	<b>16</b>
<b>Introduction</b>	<b>19</b>
<b>1 State of the Art</b>	<b>20</b>
1.1 Introduction . . . . .	20
1.2 Algorithms Overview . . . . .	21
1.2.1 Nodal Analysis Simulation Techniques . . . . .	22
1.2.2 Numerical Integration of Nonlinear Ordinary Differential Equations . . . . .	23
1.2.3 Simulation by Static Waveshaping and Digital Filter Design . . . . .	26
1.2.4 State Space Based Approach . . . . .	28
1.2.5 Nonlinear Wave Digital Filters . . . . .	31
1.2.6 Volterra Series . . . . .	34
1.3 Recent Advances . . . . .	37
1.3.1 Advances in State-Space Modeling . . . . .	38
1.3.2 State-Space Approach for Parametric Circuits . . . . .	40
1.4 Basic Circuit Component Models for Real-time Audio Effect Simulation . . . . .	42
1.4.1 Discretized Models of Capacitor and Inductor . . . . .	43
1.4.2 Diode Model . . . . .	44
1.4.3 Transistor Model . . . . .	44
1.4.4 Tube Models . . . . .	45
1.4.5 Transformer Core Models . . . . .	46
<b>2 Goals of Thesis</b>	<b>48</b>
<b>3 Circuit Analysis of Audio Effects</b>	<b>50</b>
3.1 Simulation of Circuits with Audio Transformer . . . . .	50
3.1.1 Transformer Model . . . . .	51
3.1.2 Basic Input Stage with Transformer . . . . .	55
3.1.3 Push-Pull Tube Amplifier . . . . .	62
3.1.4 Automated Incorporation of Transformer Model into DK-method . . . . .	66
3.2 Simulation of Circuits with Operational Amplifier . . . . .	72
3.2.1 Incorporation of Operational Amplifier Model into Automated DK-method . . . . .	74
3.3 Further Considerations Regarding DK-Method . . . . .	81

3.4	Summary . . . . .	85
<b>4</b>	<b>Approximation of Implicit Nonlinear Circuit Equations</b>	<b>86</b>
4.1	Precomputation of Nonlinear Systems . . . . .	86
4.1.1	Precomputation for approximation of nonlinear ODEs . . . . .	87
4.1.2	Precomputation for approximation of the state-space nonlinearity	89
4.2	Brief Overview of Function Approximation Techniques . . . . .	91
4.3	Implementation and comparison of approximation of 1-D function . . .	91
4.3.1	Non-uniform Grid Interpolation . . . . .	93
4.4	Approximation of N-D function . . . . .	99
4.4.1	Non-uniform grid interpolation . . . . .	103
4.4.2	Parallel evaluation of interpolations . . . . .	104
4.5	Customized approximation of transfer function . . . . .	106
4.5.1	Reshaping of transfer function . . . . .	106
4.6	Summary . . . . .	110
<b>5</b>	<b>Complex System Simulation</b>	<b>111</b>
5.1	Modified Block-Wise Method . . . . .	111
5.2	Guitar Tube Amplifier Simulation as a Case Study for Modified Block- wise Method . . . . .	117
5.2.1	Computational Complexity . . . . .	123
5.2.2	Simulation Results . . . . .	126
5.3	Decomposition of the DK-method nonlinear core . . . . .	129
5.3.1	Precomputation . . . . .	130
5.3.2	Further Look-up Table Size Reduction . . . . .	131
5.3.3	Simulation Results . . . . .	135
5.4	DK-model Decomposition Using Connection Components . . . . .	138
5.5	Simulation of Circuit with Global Feedback . . . . .	145
5.6	Summary . . . . .	150
<b>6</b>	<b>Quality of Simulation of Audio Effect Circuits</b>	<b>152</b>
6.1	Simulation of the Guitar Tube Preamp Engl E530 . . . . .	152
6.2	Subjective evaluation of guitar tube preamp simulation . . . . .	159
6.2.1	Results . . . . .	161
6.3	Subjective Comparison of Interpolation Techniques . . . . .	162
6.4	Subjective Comparison of Output Transformer Model . . . . .	163
6.5	Audible Aliasing Distortion . . . . .	163
6.6	Summary . . . . .	165
<b>7</b>	<b>Conclusion</b>	<b>166</b>



Author's Publications	169
Bibliography	171
List of appendices	179
A Interpolation Techniques Comparison	180
B Implementation of Interpolation Formulas	182
C Incidence Matrices for Fender Type Preamp	184
D Incidence Matrices for Marshall Preamp	185
E Incidence Matrices for Marshall Preamp with the Decomposition	187
F K matrix for Marshall Preamp with the Decomposition	189
G Instructions for Listening Tests	190
H Answer Form for Listening Tests	192
I DVD content	195

# LIST OF FIGURES

1.1	Block diagrams of Wiener (top), Hammerstein (middle) and Wiener-Hammerstein nonlinear model. . . . .	35
2.1	Block diagram of guitar amplifier . . . . .	48
3.1	Model of a transformer with two windings. . . . .	51
3.2	Magnetic part of gyrator-capacitor transformer model. . . . .	53
3.3	A transformer connected with a symmetrical voltage signal source. . . . .	55
3.4	Output signal of the input stage circuit. . . . .	59
3.5	Hysteresis loop of transformer models used in input stage circuit. . . . .	59
3.6	Solution of input stage nonlinear function with Frohlich model. . . . .	61
3.7	Solution of input stage nonlinear function with the GC model. . . . .	62
3.8	Circuit schematic for push-pull power tubes part of the tube power amplifier. . . . .	63
3.9	Output signals of the push-pull amplifier for different transformer models. . . . .	64
3.10	Distortion analysis of the push-pull amplifier with the linear model of the output transformer. . . . .	65
3.11	Distortion analysis of the push-pull amplifier with the nonlinear model of the output transformer. . . . .	65
3.12	Stamp of linear transformer model for conductance matrix . . . . .	68
3.13	Stamp of nonlinear transformer model for the conductance matrix . . . . .	71
3.14	Circuit schematic for the inverting amplifier (left) and comparator (right). . . . .	74
3.15	Transfer function of the inverting amplifier with model of OPA for real-time processing. . . . .	74
3.16	Transfer function of the inverting comparator with model of OPA for real-time processing. . . . .	75
3.17	Solution of the nonlinear equation for the comparator circuit using DK-method. . . . .	78
3.18	Solution of the nonlinear equation for the comparator circuit using equation (3.115). . . . .	78
3.19	Solution of the nonlinear equation for the inverting amplifier. . . . .	79
3.20	Circuit schematic of the simple LFO generator. . . . .	80
3.21	Transient analysis of the LFO generator simulated with the DK-method. . . . .	81
4.1	Circuit schematic for triode tube amplifier. . . . .	88
4.2	Computational cost of interpolation algorithms. Measured for one million of interpolations. . . . .	94
4.3	Circuit schematic of the nonlinear core of the Distortion effect. . . . .	95
4.4	Approximation of the transfer function of the diode clipper circuit. . . . .	97

4.5	Error of the approximations of the transfer function of the diode clipper circuit. . . . .	98
4.6	Average error of the approximations of the transfer function of the diode clipper circuit. . . . .	98
4.7	Number of operations required by $N$ -dimensional interpolation. . . . .	101
4.8	Number of coefficients to be stored for $N$ -dimensional interpolation. Spline interpolation 2 and 3 are the same as well as Linear, Hermite and Newton are the same. . . . .	101
4.9	Computation time of one million $N$ -D interpolations. . . . .	102
4.10	The transfer function of the tube preamp . . . . .	107
4.11	Reshaping coefficients dependency on cathode capacitor voltage . . . . .	109
4.12	Reshaped transfer function of the tube preamp. . . . .	109
5.1	Example of decomposition into separate blocks using the modified block-wise method. . . . .	112
5.2	Circuit schematic for typical tube amplifier stage. . . . .	113
5.3	Input volt-ampere characteristics (top) and resistance (bottom) of tube amplifier stage. . . . .	114
5.4	Transfer function of the tube amplifier stage with different load resistor. . . . .	114
5.5	Transient analysis of the tube amplifier stages with the nonlinear load and the constant load. . . . .	115
5.6	Comparison of the output signals for different load resistor in the tube stage connected as the nonlinear load. . . . .	116
5.7	Comparison of the output signals of the first tube stage with one and two tube stages connected in series as the nonlinear load. . . . .	117
5.8	Circuit schematic for the guitar preamp with four tube stages and its decomposition into three blocks. . . . .	118
5.9	Circuit schematic of the block 1 of the decomposed tube preamp. . . . .	120
5.10	Circuit schematic of the block 1 and 2 of the decomposed tube preamp. . . . .	120
5.11	Approximating functions for the simulation of the system (5.1) . . . . .	121
5.12	Approximating functions for the simulation of the system (5.1). . . . .	122
5.13	Simulation results for a logarithmic sweep signal. The plate voltage signals of all tubes are displayed. . . . .	127
5.14	Comparison between simulation results using the numerical solution and using approximations for a part of the real guitar riff. Only the error signals are displayed. . . . .	128
5.15	Circuit schematic of the Fender type guitar preamp. . . . .	130
5.16	Output signals (top, dashed line for the numerical solution) and the difference between numerical and approximated solution in the time domain without the parameter interpolation. . . . .	136

5.17	Difference between the numerical and approximated solution in the time frequency without the parameter interpolation. The numerical solution (dashed line) is shifted to the right. . . . .	137
5.18	Output signals (top, dashed line for the numerical solution) and the difference between the numerical and approximated solution in the time domain with the parameter interpolation. . . . .	137
5.19	Difference between the numerical and approximated solution in the time frequency with the parameter interpolation. The numerical solution (dashed line) is shifted to the right. . . . .	138
5.20	Circuit schematic of the Marshall JCM 800 guitar preamp. . . . .	139
5.21	Circuit schematic of the Marshall JCM 800 guitar preamp with the decomposition into blocks. . . . .	141
5.22	Output signals (top) and their difference (bottom) for the simulation of Marshall preamp with and without the decomposition . . . . .	145
5.23	Circuit schematic of the tube power amplifier. . . . .	147
5.24	Frequency response magnitude of the first harmonic component for different topologies of the circuit for parameter "Presence" set to zero. . . . .	149
5.25	Frequency response magnitude of the first harmonic component for different topologies of the circuit for parameter "Presence" set to one. . . . .	149
5.26	Frequency response magnitude of measured loudspeaker cabinets – the Mesa Boogie Rectifier and the Marshall JCM 800. . . . .	150
6.1	Comparison of measured and simulated voltage signals at plate terminals of the tubes for input sine wave with the amplitude 0.5 V. . . . .	153
6.2	Comparison of measured and simulated voltage signals at plate terminals of the tubes for input sine wave with the amplitude 2 V. . . . .	154
6.3	Comparison of measured and simulated frequency responses (for the first harmonic content). . . . .	154
6.4	Spectrogram of the output signal - measured clean channel. . . . .	155
6.5	Spectrogram of the output signal - simulated clean channel. . . . .	155
6.6	Comparison of measured and simulated voltage signals at plate terminals of the tubes for input sine wave with the amplitude 0.5 V. . . . .	156
6.7	Comparison of measured and simulated voltage signals at plate terminals of the tubes for input sine wave with the amplitude 2 V. . . . .	157
6.8	Comparison of measured and simulated frequency responses (for the first harmonic content). . . . .	157
6.9	Spectrogram of the output signal - measured crunch channel. . . . .	158
6.10	Spectrogram of the output signal - simulated crunch channel. . . . .	158

# LIST OF TABLES

3.1	Values of circuit components of input stage circuit. . . . .	57
3.2	Parameter of Frohlich and GC-model of transformer core. . . . .	57
3.3	Parameter of JA-model of transformer core. . . . .	58
3.4	Number of iterations of Newton-Raphson method. . . . .	58
3.5	Values for circuit elements for push-pull power tubes part of the tube power amplifier. . . . .	63
3.6	Parameter of Frohlich and GC-model of the transformer core for the transformer Fender NSC041318. . . . .	63
3.7	Values of circuit components for the LFO generator. . . . .	80
4.1	Computational cost comparison of interpolation algorithms. . . . .	93
4.2	Element values for schematic on Figure 4.3 . . . . .	96
4.3	Comparison of the number of data points required for approximation of diode clipper circuit. . . . .	97
4.4	Maximal number of interpolations per sample for $f_s = 48$ kHz. . . . .	102
4.5	Maximal number of interpolations per sample for $f_s = 192$ kHz. . . . .	103
5.1	Values for circuit components for typical tube amplifier stage. . . . .	113
5.2	Values for circuit components for the guitar preamp. . . . .	118
5.3	Look-up table for simulation of the first block. . . . .	122
5.4	Look-up table for simulation of the second and third block. . . . .	123
5.5	Computational complexity comparison of the simulations based on the Newton method – the number of operations. . . . .	126
5.6	Computational complexity comparison of the simulations based on approximations – the number of operations. . . . .	126
5.7	Errors for the simulation from Figure 5.14. The plate voltage signal errors are displayed. . . . .	129
5.8	Component values of the Fender type guitar preamp . . . . .	129
5.9	Ranges of $\mathbf{p}$ parameters. . . . .	131
5.10	Ranges of $\mathbf{K}$ coefficients. . . . .	131
5.11	Covariance between precomputed functions and inputs. . . . .	132
5.12	Description of the look-up tables – number of intervals . . . . .	135
5.13	Data size of the look-up tables. . . . .	135
5.14	Component values of the Marshall JCM 800 guitar preamp . . . . .	139
5.15	Component values of the tube power amplifier. . . . .	146
6.1	Results of subjective evaluation of preamp simulation. . . . .	161
6.2	Results of the comparison of interpolation techniques using the listen- ing test. . . . .	163
6.3	Results of transformer model listening test. . . . .	164

6.4 Results of audible aliasing distortion listening test. . . . . 164

# LIST OF ABBREVIATIONS

ASIO	Audio Stream Input/Output
BCT	Binary Connection Tree
BE	Backward Euler Formula
BJT	Bipolar Junction Transistor
DC	Direct Current
DSP	Digital Signal Processing
FE	Forward Euler Formula
FIR	Finite Impulse Response
GC	Gyrator-Capacitor
GUI	Graphical User Interface
IIR	Infinite Impulse Response
JA	Jiles-Atherton
JFET	Junction Field Effect Transistor
KCL	Kirchhoff's Current Law
KVL	Kirchhoff's Voltage Law
LMS	Least Mean Square
LHS	Left Hand Side
LFO	Low Frequency Oscillator
MMF	Magneto-Motive Force
MNA	Modified Nodal Analysis
ODE	Ordinary Differential Equations
OPA	Operational Amplifier
PCA	Principal Component Analysis
RFP	Reflection Free Port

RHS Right Hand Side

RK Runge-Kutta

SIMD Single Instruction Multiple Data

SPICE Simulation Program with Integrated Circuit Emphasis

SSE Streaming SIMD Extensions

SVD Singular Value Decomposition

THD Total Harmonic Distortion

TR Trapezoidal Rule

VST Virtual Studio Technology

WDF Wave Digital Filter



# LIST OF SYMBOLS AND MATH OPERATIONS

$\mathbf{A}, \mathbf{B}, \mathbf{C}$	DK-method matrix
$\mathbf{A}_o, \mathbf{B}_o, \mathbf{C}_o$	K-method matrix
$B$	magnetic flux density
$B_{\text{SAT}}$	magnetic flux density in saturation
$C$	capacitance
$\mathbf{D}, \mathbf{E}, \mathbf{F}$	DK-method matrix
$f_s$	sampling frequency
$\Phi$	magnetic flux
$\Phi_{\text{SAT}}$	magnetic flux in saturation
$\mathbf{G}, \mathbf{H}, \mathbf{K}$	DK-method matrix
$\mathbf{G}$	conductance matrix
$\mathbf{G}_r$	resistor conductance matrix
$\mathbf{G}_x$	energy-storing component conductance matrix
$G$	conductance
$H$	magnetic force
$h_N$	$N$ -dimensional impulse response (Volterra kernel)
$f()$	nonlinear function
$\mathcal{F}$	magnetomotive force
$i, \mathbf{i}$	current, vector of currents
$i$	index
$\mathbf{I}$	identity matrix
$i_B$	base current
$i_C$	capacitor current, collector current
$i_E$	emitor current
$i_g$	grid current
$i_L$	inductor current
$i_p$	plate current
$i_S$	saturation current

$i_s$	screen current
$i_{\text{conn}}$	connection current
<b>J</b>	Jacobian matrix
$L$	inductance
$l_{\text{mag}}$	length of magnetic path
$M$	magnetization
$\mu$	permeability, tube amplification factor
<b>M</b>	conductance matrix
$\mathbf{M}_x$	conductance matrix for state variables
$\mathbf{M}_u$	conductance matrix for input variables
$\mathbf{M}_n$	conductance matrix for nonlinear components
$\mathbf{N}_n$	nonlinear component incidence matrix
$\mathbf{N}_o$	outputs incidence matrix
$\mathbf{N}_{\text{OPA}}$	operational amplifier incidence matrix
$\mathbf{N}_r$	resistor incidence matrix
$\mathbf{N}_t$	transformer incidence matrix
$\mathbf{N}_u$	inputs incidence matrix
$\mathbf{N}_v$	variable component incidence matrix
$\mathbf{N}_x$	energy-storing component incidence matrix
$O()$	computational complexity
$p$	complex Laplace parameter
<b>p</b>	input variable for DK-method
$\mathbb{R}^N$	$N$ -dimensional Euclidean space
$R$	resistance
$\mathcal{R}$	reluctance
$S$	cross-section area
<b>S</b>	conductance matrix
$\mathbf{S}_{\text{OPA}}$	conductance matrix with operational amplifier model
$\mathbf{S}_t$	conductance matrix with transformer model

$T$	sampling period
$x$	input variable, state variable
$\mathbf{u}$	input vector
$\mathbf{x}$	state vector
$\dot{\mathbf{x}}$	time derivation of state vector
$v, \mathbf{v}$	voltage, vector of voltages
$v_T$	thermal voltage
$z$	complex parameter in $\mathbb{Z}$ domain

# INTRODUCTION

Digital signal processing influences many fields of human interests these days and together with the progress of computer science it addresses many of new applications. One of these applications is digital audio signal processing in computers. Original analog records on magnetic tapes have been replaced with digital records on hard disk drives in recording studios. Furthermore, an additional processing of audio signals by audio effects, such as compressors and equalizers, is replaced by digital signal processing using real-time digital audio effects. Basically, the digital audio effects work similarly as their analog prototypes, but the output audio signal usually differs. Some musicians claim that the digital audio effects sound too accurate and thin, and therefore they prefer the analog audio effects. Because of this, there is a big effort to simulate these analog audio effects digitally, including their imperfections, such as nonlinear distortion of audio signals, nowadays. A special category of these audio effects is built by simulations of guitar amplifiers and effects where nonlinear distortion is required and the aim is to get the same nonlinear distortion like the analog prototypes have.

These new technologies can be used in recording studios, either home or professional, but they are also part of audio multi-effects for live performance and even guitar amplifiers with digital signal processors can be found on the market. The biggest advantage is universality of these systems – simulations of several analog devices, which are often large and expensive, can be inside one system. The only one disadvantage is that the simulation is still insufficiently accurate, but this improves with the new algorithms and more powerful processors.

Simulated devices can be usually split into two categories. The first one contains linear systems and analog devices which have very small amount of nonlinear distortion. To simulate these devices, quite simple and common techniques, such digital filters designed according to analog prototypes, can be used.

The other category contains strongly nonlinear systems. To analyze and simulate this type of systems, more complex techniques, which often have enormous computational demands, must be used. Therefore, there is effort to find such algorithms that will lead to the same audio perception as the simulated nonlinear system, while the computational demands will stay relatively low to be workable in real-time.

# 1 STATE OF THE ART

An overview of algorithms for simulation of audio effects will be given in this chapter. Basic circuit components used for these simulations will be described as well. Finally, recent advances in improvement of the algorithms will be mentioned.

## 1.1 Introduction

Development of algorithms for simulation of analog electronic circuits has begun with computer programs for analysis of the electronic circuits. Program Simulation Program with Integrated Circuit Emphasis (SPICE) [13] is a typical example of such programs. This program and its derivations have been developing since 1973. They have undergone many improvements and optimizations since then. However, the main algorithm, which is analysis of circuit nodes and numerical solution of circuit equations, remains the same. The time-domain simulation of nonlinear audio effects here is related to the transient analysis of given circuits. These programs nowadays allow simulation of majority of electronic circuits (analog, digital, audio circuits, power circuits, etc.), but this generality of algorithm has enormous computational demands and thus, they are not suitable for real-time simulations.

In the case of audio effects, the generality of the simulation algorithm is not necessary. In spite of generality, attention is rather paid to several typical circuits which can be simplified, approximated and optimized without losing the same audio perception. Therefore, it is important to understand how the analog audio effects work in order to design proper simplifications (e.g. division into several separate blocks according to their function). The simplified blocks can be then simulated using different algorithms exploiting benefits which different algorithms provide.

A block that implements a nonlinear function is the basic block of all nonlinear algorithms. This nonlinear block can be implemented as memoryless (static) or with memory (dynamic). Both cases produce higher harmonics which spread the signal spectrum [14]. A digital signal which originally complied Nyquist theorem now contains higher harmonics which do not comply Nyquist theorem anymore, which results in generation of aliasing distortion. Some algorithms are designed in such way that aliasing distortion can be removed by connecting a low-pass filter before the block with the nonlinear function which will filter out all higher frequencies which would cause aliasing distortion [14]. Generally, it is possible to reduce aliasing distortion using the over-sampling technique – the input signal is several times up-sampled, then processed by the nonlinear function and finally down-sampled [14, 15]. But even this technique will not remove aliasing distortion completely, because when one simulates extreme nonlinearities, where Fourier series goes to infinitive, the

over-sampling factor would have to be very high and so would be computational demands. In this case, one has to make use of psychoacoustics and design the over-sampling factor in such way that aliasing distortion will be masked with the masking threshold. This approach requires thorough analysis of the algorithm for given set of input signals and the over-sampling factor can be designed on the basis of listening test or psychoacoustic models of masking. Attempts to analyze aliasing distortion using the masking effect can be found in [16, 17].

## 1.2 Algorithms Overview

Generally, the algorithms for simulation of analog audio effects can be classified into several categories – a black box approach or a white box approach, with linear or nonlinear transfer function, etc. Good overview can be found e.g. in literature [18]. The black box or white box approach differ in the fact whether the inner structure of simulated systems is known. In case of the black box approach (uninformed), the inner structure is unknown and all the necessary information for designing the simulating system is extracted only from relation between the input and output signal. This approach is often used with linear algorithms that are foremost based on digital linear filters and the only task here is to determine filter coefficients, which can be done by linear system identification task using e.g. Least Mean Square (LMS) algorithm or by measuring the impulse response of the unknown system, which enables direct derivation of Finite Impulse Response (FIR) filter coefficients. Further, it is possible to reduce computational costs by approximation of the FIR filter frequency response by Infinite Impulse Response (IIR) filter. However, the black box approach can be used also for simulation of nonlinear systems. In this case, the core of the simulation algorithm is usually built with Volterra series equation and the unknown is a set of coefficients for this nonlinear filter. Further information are mentioned in chapter 1.2.6.

Although the black box approach offers universal and accurate methods for simulation of analog audio systems, its use is limited, especially for circuits with variable circuits components, because simulation of such systems requires measuring and storing data for different values of parameters which results in large look-up tables. Even though it can be solved by linear interpolations e.g. as in [19], white box methods are more convenient in this case.

The basic feature of the white box methods is that they know the inner structure of the simulated system. Talking about the analog audio effects, the inner structure of the system is directly the circuit schematic of the system and thus these types of algorithms are known as circuit based simulation techniques where individual

algorithms differ in processing of the circuit schematic. This type of algorithms is aim of this thesis. Therefore, these algorithms will be further described in following text.

### 1.2.1 Nodal Analysis Simulation Techniques

The analysis of electronic circuits makes use of Kirchhoff's laws: Kirchhoff's Current Law (KCL) and Kirchhoff's Voltage Law (KVL). There are several techniques for the analysis of electronic circuits, but the loop analysis of electronic circuits with unknown loop currents and the nodal analysis of electronic circuits with unknown nodal voltages are the most often used techniques for the analysis of complex systems. Both of them are suitable for different types of electronic circuits. Although the loop analysis is more straightforward and convenient for simple circuits, it requires the use of graph theory for determining loops in more complicated circuit (the loops must be independent, otherwise it results in overdetermined system) and therefore it is not suitable for automated circuit analysis with computer aided systems. On the contrary, the nodal analysis requires determination of unknown independent nodal voltages and one reference node, which is the ground node, and this procedure can be easily automated and therefore many of electronic circuit simulators are based on this method. The nodal analysis uses the KCL

$$0 = \sum i_n \quad (1.1)$$

for each node  $n$  except the reference, which results in the matrix system description

$$\mathbf{G}\mathbf{v} = \mathbf{i} \quad (1.2)$$

with current input sources  $\mathbf{i}$  on the Right Hand Side (RHS), the unknown voltage vector  $\mathbf{v}$  and the conductance matrix  $\mathbf{G}$  which consists of conductances of passive circuit components and also contains linearized active circuit components. Obtaining the  $\mathbf{G}$  matrix is possible using so called stamps of the circuit components and it is not further addressed here, since it can be found in various literature [20, 21, 22, 23]. However, it should be noticed here that models of accumulating components have to be discretized in time-domain since the time-domain analysis is the main objective here. The stamps for these components can be found in [20, 21, 22] and the models will be further reviewed in chapter 1.4. Each accumulating component has then a current contribution for the RHS which corresponds to the accumulating component state.

Nonlinear circuit components, namely a diode, a triode, a transistor etc., are expressed as voltage controlled current sources placed on the RHS in form of

$$i = f(\mathbf{v}) \quad (1.3)$$

where  $i$  is the controlled current,  $f$  is the nonlinear function describing behavior of the component model and  $\mathbf{v}$  is the vector of controlling voltages.

The real circuit sources are the last contribution to the sources on the RHS. However, the nodal analysis can handle only the ideal current sources, while the audio effect circuits usually contain ideal voltage sources. Therefore the Modified Nodal Analysis (MNA) is used instead of the nodal analysis method. The MNA extends the system (1.2) by unknown currents of ideal voltage sources on the Left Hand Side (LHS) while input voltages are placed on the RHS. The resulting equation describing the discretized nonlinear system in the matrix form is according to [20, 22]

$$\mathbf{M}\mathbf{v}[n] = \mathbf{M}_x\mathbf{x}[n-1] + \mathbf{M}_u\mathbf{u}[n] + \mathbf{M}_n\mathbf{i}(\mathbf{v})[n] \quad (1.4)$$

where  $\mathbf{M}$ ,  $\mathbf{M}_x$ ,  $\mathbf{M}_u$  and  $\mathbf{M}_n$  are conductance matrices of the MNA method,  $\mathbf{x}$  is the state of the system in the previous sample period,  $\mathbf{u}$  is the vector of inputs (currents or voltages) and  $\mathbf{v}$  is the vector of unknown nodal voltages and currents of ideal voltage sources,  $\mathbf{i}$  is the vector of nonlinear currents and  $n$  denotes the time index. Solving of equation (1.4) leads to numerical solving with the residual function

$$\mathbf{0} = \mathbf{M}_x\mathbf{x}[n] + \mathbf{M}_u\mathbf{u}[n] + \mathbf{M}_n\mathbf{i}(\mathbf{v})[n] - \mathbf{M}\mathbf{v}[n] \quad (1.5)$$

and after solving to state update by

$$\mathbf{x}[n+1] = \mathbf{M}_x^{-1}(-\mathbf{M}_u\mathbf{u}[n] - \mathbf{M}_n\mathbf{i}(\mathbf{v})[n] + \mathbf{M}\mathbf{v}[n]). \quad (1.6)$$

Using this approach and accurate models of components, the great accuracy of the simulation can be achieved. However, since the equation (1.5) is in implicit form and moreover, sizes of matrices  $\mathbf{M}$ ,  $\mathbf{M}_x$ ,  $\mathbf{M}_u$  and  $\mathbf{M}_n$  depend on the number of circuit nodes, this approach is not suitable for real-time audio processing due to high computational costs. However, this algorithm can serve as a baseline to which all algorithms working in real-time should converge.

## 1.2.2 Numerical Integration of Nonlinear Ordinary Differential Equations

Numerical integration of nonlinear Ordinary Differential Equations (ODE) is closely related to the previous chapter because the equation (1.4) is essentially an ODE which has already been discretized using Backward Euler Formula (BE) or Trapezoidal Rule (TR) (see later in this chapter and chapter 1.4) where component-wise discretization has been performed.

The numerical integration of ODE is a widespread known problem and has been aimed in various literature over the last three decades and therefore, it is not



addressed here. However, the usage of numerical integration for real-time simulation of audio effects was studied intensively in literature [24, 25, 22] with focus on different numerical integration formulas: implicit integration formulas, represented by BE of the first and second order and TR, which all require a numerical algorithm to solve the nonlinear equation because these formulas generate delay free loops, further semi-implicit methods which require one iteration of the numerical algorithm to solve the nonlinear equation and which were represented by semi-implicit BE formula and finally explicit methods represented by Forward Euler Formula (FE) and Runge-Kutta (RK) method of the fourth order which do not require the use of the numerical algorithm to solve the nonlinear equation because these formulas depend only on previous system states.

The integration formulas were tested in simulation of a diode clipper circuit, which is the important part of many guitar distortion effect pedals. To apply different integration formulas, the nonlinear circuit equation should be in form

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) \quad (1.7)$$

where  $\mathbf{x}$  is the state vector and  $\mathbf{u}$  is the input vector and  $\dot{\mathbf{x}} = \frac{d\mathbf{x}}{dt}$ . Various input signals were used to investigate computational cost, accuracy of the solution as well as stability of the formulas. It was found that, although the explicit formulas are stable for some types of ODEs like in [26], the implicit formulas must be generally used in order to ensure the stability of the solution. This was examined in [22] where there is stated that the solution stability of explicit formulas is related to the ratio between largest values of eigenvalues of (1.7) and the sampling frequency value. Therefore, for the given sampling frequency, there is a limit on the largest value of negative eigenvalue which depends on operating point and therefore these formulas are not stable in the whole range of input conditions unless the function  $\mathbf{f}(\mathbf{x}, \mathbf{u})$  from (1.7) is compressive, which is the case of [26]. On the contrary, implicit formulas place no limits on the maximum of values of eigenvalues and therefore they can work with low sampling frequencies. If there is no constrain on the value of the sampling frequency for the bounded solution, then such method can be denoted as stiff stable. It implies that explicit formulas are not stiff stable.

Computational costs and accuracy of the solution are thus closely related to the stability of the solution. Although explicit formulas offer constant and low computational cost, the requirement for enormous sampling frequencies makes them unpractical for usage in simulations of electronic circuits in real-time. Computational cost of implicit formulas is related to the number of iterations of a numerical nonlinear solver (e.g. Newton-Raphson method) and differs for different input signals. Lower computational costs can be usually obtained for signals with low frequency content. To decrease the difference in computational cost for signals with the low frequency

content and the high frequency content, the oversampling can be applied which further reduces aliasing distortion. Implicit formulas are capable of working in real-time for some simple audio effect circuits. However the computational cost is not spread equally and some peaks of the computational cost can cause drop-outs of the output signal if the computational power of the computational system is limited. Further, although implicit methods are stable, the stability of the solution also depends on the stability of the used numerical nonlinear solver. The commonly used Newton-Raphson solver converges very fast if the initial solution is close to the correct solution, otherwise it can diverge. The homotopy method was developed to cope with this problem. It was originally used in paper [27] to ensure a global convergence for finding of DC operating point of a nonlinear circuit and further it was used in [22] to ensure a global convergence of the Newton-Raphson method. However, this introduces additional computational cost of the algorithm. The semi-implicit methods seem to be trade-off providing constant computational cost and stability, however according to [22], they introduce artificial sound artifacts at high frequencies.

Based on the comparison of integration formulas made in [22], it is possible to state that BE and TR are applicable for real-time simulations if sufficient computational power is available. Both methods can also be exploited during the component-wise discretization of circuits components and therefore they will be mentioned here. The BE formula is given by

$$x[n] = x[n - 1] + T\dot{x}[n] \quad (1.8)$$

where  $T$  is the sampling period and  $\dot{x}$  is time derivative of  $x$ . In terms of digital signal processing, the time derivative can be expressed as

$$\dot{x}[n] = \frac{1 - z^{-1}}{T}x[n] \quad (1.9)$$

which is Backward Euler discretization formula. The TR integration formula is given by

$$x[n] = x[n - 1] + \frac{T}{2}(\dot{x}[n] + \dot{x}[n - 1]) \quad (1.10)$$

and TR discretization formula is then

$$\dot{x}[n] = \frac{2}{T} \frac{1 - z^{-1}}{1 + z^{-1}}x[n] \quad (1.11)$$

that is closely related to Bilinear transform, often used for discretization of an analog transfer function to a digital one. To compare both methods, the BE provides good frequency warping properties but introduces numerical damping which can affect frequency response, while TR does not introduce artificial damping, however there is frequency warping due to mapping of infinite  $s$ -plane to  $z$ -plane unit circle [28, 22].

### 1.2.3 Simulation by Static Waveshaping and Digital Filter Design

Algorithms based on static waveshaping on the contrary to the numerical integration minimize overall computation costs. This type of algorithms was designed in correspondence with low computational power of computational systems and even today they can find their employment in Digital Signal Processing (DSP) applications. The detailed overview of these algorithms can be found in [18].

A nonlinear behavior of algorithms is specified by a simple mapping between input and output signals given by

$$y[n] = f(x[n]) \quad (1.12)$$

where  $f(x)$  is the nonlinear transfer function. Since this function is time-invariant, these types of algorithms are called static waveshapers. The shape of the transfer function  $f(x)$  depends on the particular circuit to be solved (used circuit components and topology). The transfer function can be obtained by a numerical solution of the given circuit, from output data of programs for electronic circuits simulation or by a measurement of the real circuit.

The numerical solution of the circuit usually gives the best results, however, it is necessary to have sufficiently accurate models of nonlinear circuit components as has already been stated in previous chapters. In order to obtain the function  $f(x)$ , the circuit has to be solved for different input signal values and then the transfer function is obtained from the input-output signal relation. Because the circuits usually contain accumulative components, it requires DC solution of the given circuit. This can be done by omitting accumulative components or by setting the state derivatives in circuit equations to zero as mentioned in [22]. Determination of the transfer function  $f(x)$  from output data of programs for electronic circuits simulation is discussed in [29]. A decaying sine burst with alternating polarity was fed at the input of the investigated circuit. Samples of the input and output signals are then sorted according to their values in order to get the transfer function. The same way can also be used when a real circuit is measured. However, by this measurement, the transfer function is obtained only for harmonic stable state and therefore, the results are valid only for circuits without accumulating components. In case of dynamic circuits, the transfer function can significantly vary for different frequencies.

If the shape of the transfer function  $f(x)$  is known, it is necessary to find such function (1.12) which would approximate the transfer function. Various functions  $f(x)$  approximating different electronic circuits can be found in literature [14, 18]. All these functions approximate one typical transfer function and they are not suitable for universal usage. If a general shape of the transfer function is desired, the shape

can be approximated by the polynomial function

$$f(x) = \sum_{i=0}^N b_i x^i \quad (1.13)$$

where coefficients  $b_i$  are Taylor polynomial coefficients [14] and they can be estimated from the Taylor series. It is also possible to determine these coefficients from the output signal spectrum [15]. This implementation is suitable for the simulation of circuits with smaller nonlinearity or exciter audio effects where it is possible to change ratio of higher harmonics. When the polynomial approximation (1.13) is used in wider range of the input signal, polynomials of very high orders have to be used. However, it is possible to approximate the transfer function by piecewise approximation. The combination of linear and polynomial functions is mentioned in [1]. The transfer function is given by

$$f(x) = \begin{cases} a & x < x_1 \\ \sum_{i=0}^N b_i x^i & x \geq x_1 \wedge x \leq x_2 \\ c & x > x_2. \end{cases} \quad (1.14)$$

Naturally, it is desired to have a continuous transfer function. If the points  $x_1, x_2$  are inflection points of the polynomial, the first order continuity is achieved. Very good results can be achieved using interpolation of data stored in look-up tables, which was used in several algorithms mentioned in [18] and further in [30, 31, 17, 20, 21, 22]. Mostly, the linear interpolation was used. The cubic spline interpolation of the transfer function was used in [2].

Nevertheless, simulated circuits often contain frequency dependent circuit components which influence the frequency response of the simulated circuit. Because the static transfer function is not capable to change the frequency response, these algorithms must be supplemented with additional digital filters connected before and behind the nonlinear function. There is again the whole variety of algorithms [18] implementing the static transfer function together with digital filters but the accuracy, which would be comparable with the numerical integration of the circuit ODEs, was not accomplished. The explanation why the simulation based on the static transfer function and digital filters is not accurate is given in [22].

Although these algorithms are not as accurate as the numerical integration, they are useful for simulation of complex systems where the numerical integration would require extreme computation power to be workable in real-time. In such case, the complex circuit of the audio effect is divided into linear and nonlinear parts and the whole system is simulated using a series of digital filters and memoryless nonlinear functions. This approach was used in many commercial products and patents [18] as well as in academic sphere [15]. The circuit-based simulation of the

complex system can be found in [22, 32] where the simulation of guitar distortion effect pedal and guitar overdrive effect pedal was discussed. The nonlinear functions were obtained from the DC solution of nonlinear circuit. The rest of circuits was divided into analog linear filters. Consequently, symbolic analysis of these filters was performed and finally, digital filters were designed using the Bilinear transform of symbolically derived analog transfer functions. This procedure is described in detail e.g in [33, 34]. The simulation of both guitar pedals was also compared with measured originals. Results showed validity of this approach although it is not the exact emulation of circuits. Computational cost is comparable to commercial products. Further accuracy can be achieved by using more accurate nonlinear devices, more sophisticated nonlinear block simulation and probably better division into blocks.

### 1.2.4 State Space Based Approach

The main problem of direct numerical integration of circuit equation from chapters 1.2.1, 1.2.2 is, except the necessity of use of a nonlinear iterative solver, total number of equations and thus number of unknown variables to be solved numerically. The nonlinear equations are spread over the unknowns forming non-computable loops – delay free loops. However, as it was showed in prior work [35] and then reviewed in [20, 21, 22], the number of unknown variables to be solved numerically can be significantly reduced by using transformation of linear parts of the circuit equations which highlights nonlinear mappings. Since this method has worked with Kirchhoff’s variables (in contrast to other methods reducing the delay free loops), it was called the K-method. The main assumption in [35] is that the continuous-time system can be decomposed into a nonlinear multiple-input multiple-output mapping part and a linear dynamic part. The circuit equations can be rewritten to

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} + \mathbf{C}\mathbf{i} \quad (1.15)$$

$$\mathbf{i} = \mathbf{f}(\mathbf{v}) \quad (1.16)$$

$$\mathbf{v} = \mathbf{D}\mathbf{x} + \mathbf{E}\mathbf{u} + \mathbf{F}\mathbf{i}, \quad (1.17)$$

where  $\mathbf{x}$  is the state variable of the system and  $\dot{\mathbf{x}}$  its time derivative,  $\mathbf{u}$  is the input of the system (the notation of variables differs here from the original one [35] because the notation from [22] was used). Equation (1.16) represents the nonlinear mapping function and the variable  $\mathbf{v}$  contains transformed circuit equations which serve as the input to the nonlinear mapping function  $f(\cdot)$ . The output variable can be expressed as

$$\mathbf{y} = \mathbf{A}_o\mathbf{x} + \mathbf{B}_o\mathbf{u} + \mathbf{C}_o\mathbf{i}. \quad (1.18)$$

To get a discrete time version, a discretization of the state variable derivative is necessary. The process of discretization for both BE and TR discretization is described in [35]. The original method was developed for the simulation of nonlinear acoustic systems and therefore, it was reviewed in [20, 21, 22] in order to adapt the method for the simulation of electronic circuits. The nonlinear electronic component models are usually specified as voltage-controlled current sources  $i = f(v)$  as it has already been mentioned earlier. This term is in correspondence with the equation (1.16) where the variable  $\mathbf{i}$  holds currents of nonlinear component models and variable  $\mathbf{v}$  holds controlling voltages on nonlinear component model terminals. A subsequent discretization, TR for instance, of equation (1.15) leads to the discrete state update equation

$$\mathbf{x}[n] = \mathbf{H}(\alpha\mathbf{I} + \mathbf{A})\mathbf{x}[n-1] + \mathbf{H}\mathbf{B}(\mathbf{u}[n] + \mathbf{u}[n-1]) + \mathbf{H}\mathbf{C}(\mathbf{i}[n] + \mathbf{i}[n-1]) \quad (1.19)$$

where  $\mathbf{H} = (\alpha\mathbf{I} - \mathbf{A})^{-1}$ ,  $\alpha = 2/T$  for bilinear transform without frequency prewarping. Further substitution of (1.19) into (1.17) results in

$$\mathbf{v}[n] = \mathbf{p}[n] + (\mathbf{D}\mathbf{H}\mathbf{C} + \mathbf{F})\mathbf{i}[n] \quad (1.20)$$

with

$$\mathbf{p}[n] = \mathbf{D}\mathbf{H}(\alpha\mathbf{I} + \mathbf{A})\mathbf{x}[n-1] + (\mathbf{D}\mathbf{H}\mathbf{B} + \mathbf{E})\mathbf{u}[n] + \mathbf{D}\mathbf{H}\mathbf{B}\mathbf{u}[n-1] + \mathbf{D}\mathbf{H}\mathbf{C}\mathbf{i}[n-1] \quad (1.21)$$

and finally the nonlinear function can be expressed as

$$\mathbf{i}[n] = \mathbf{f}(\mathbf{K}\mathbf{i}[n] + \mathbf{p}[n]) \quad (1.22)$$

where

$$\mathbf{K} = \mathbf{D}\mathbf{H}\mathbf{C} + \mathbf{F} \quad (1.23)$$

in implicit form which can be given by a look-up table or solved numerically in real-time. The final computation scheme consist of:

1. computation of input variable  $\mathbf{p}[n]$  from equation (1.21),
2. solving the nonlinear function (1.22),
3. and update of state variable according to (1.19).

The equation (1.22) can be solved by Newton-Raphson method given, for this case, by

$$\mathbf{i}_{it+1} = \mathbf{i}_{it} - \mathbf{J}(\mathbf{i}_{it})^{-1}\mathbf{f}(\mathbf{i}_{it}) \quad (1.24)$$

with the subscript  $it$  denoting iterations, the residual function

$$\mathbf{f}(\mathbf{i}_{it}) = -\mathbf{i}_{it} + \mathbf{f}(\mathbf{K}\mathbf{i}_{it} + \mathbf{p}[n]) \quad (1.25)$$

and Jacobian

$$\mathbf{J}(\mathbf{i}_{it}) = \mathbf{J}_F(\mathbf{K}\mathbf{i}_{it} + \mathbf{p}[n]) - \mathbf{I} \quad (1.26)$$

where  $\mathbf{J}_F(\cdot)$  is the Jacobian of nonlinear components and  $\mathbf{I}$  the identity matrix. The global convergence can be ensured by using the homotopy. Details can be found in [22]. Because the unknowns of the nonlinear function are currents of the nonlinear components, the method was denoted as iteration on currents.

However, it was found in [22] that it is possible to derive the method to iterate on unknown voltages (VK-method) that has faster convergence than iteration on currents. This is due to the compressive character of the residual function of VK-method. In this case, the nonlinear component equation (1.16) is substituted into (1.17) resulting in

$$\mathbf{v} = \mathbf{D}\mathbf{x} + \mathbf{E}\mathbf{u} + \mathbf{F}\mathbf{f}(\mathbf{v}). \quad (1.27)$$

The nonlinear equation to be solved is now

$$\mathbf{v}[n] = \mathbf{p}[n] + \mathbf{K}\mathbf{f}(\mathbf{v}[n]). \quad (1.28)$$

The residual function for the Newton-Raphson method is

$$\mathbf{f}(\mathbf{v}_{it}) = -\mathbf{v}_{it} + \mathbf{p}[n] + \mathbf{K}\mathbf{f}(\mathbf{v}_{it}) \quad (1.29)$$

and Jacobian

$$\mathbf{J}(\mathbf{v}_{it}) = \mathbf{K}\mathbf{J}_F(\mathbf{v}_{it}) - \mathbf{I} \quad (1.30)$$

and the Newton-Raphson method is then

$$\mathbf{v}_{it+1} = \mathbf{v}_{it} - \mathbf{J}(\mathbf{v}_{it})^{-1}\mathbf{f}(\mathbf{v}_{it}) \quad (1.31)$$

with unknown controlling voltages on nonlinear component terminals.

Yeh in his work [20, 21, 22] also proposed the method for automated derivation of K-method matrices from the MNA description of the circuit given by equation (1.4). This method requires a prior discretization of components, as it has already been considered in (1.4), and therefore it is denoted as the DK-method. The nodal voltages from (1.4) can be computed using

$$\mathbf{v}[n] = \mathbf{M}^{-1}\mathbf{M}_x\mathbf{x}[n-1] + \mathbf{M}^{-1}\mathbf{M}_u\mathbf{u}[n] + \mathbf{M}^{-1}\mathbf{M}_n\mathbf{i}(\mathbf{v})[n] \quad (1.32)$$

while all the nodal voltages in terms of the K-method can be computed from

$$\mathbf{v}[n] = \mathbf{A}'_x\mathbf{x}[n-1] + \mathbf{B}'\mathbf{u}[n] + \mathbf{C}'\mathbf{i}(\mathbf{v})[n]. \quad (1.33)$$

The relation is obvious and resulting in  $\mathbf{A}' = \mathbf{M}^{-1}\mathbf{M}_x$ ,  $\mathbf{B}' = \mathbf{M}^{-1}\mathbf{M}_u$  and  $\mathbf{C}' = \mathbf{M}^{-1}\mathbf{M}_n$ . However, to form the matrices  $\mathbf{A}$ ,  $\mathbf{B}$ ,  $\mathbf{C}$ ,  $\mathbf{D}$ ,  $\mathbf{E}$ ,  $\mathbf{F}$ ,  $\mathbf{A}_o$ ,  $\mathbf{B}_o$ ,  $\mathbf{C}_o$ , corresponding voltages from the vector  $\mathbf{v}$  must be subtracted. This can be done by subtracting pairs of corresponding rows in matrices  $\mathbf{A}'$ ,  $\mathbf{B}'$ ,  $\mathbf{C}'$ . Further details can be found in [20].

Regarding computational complexity, the algorithm has quadratic complexity which can be expressed as  $O(N^2 + NM + NP)$  where  $N$  is the number of states,  $M$  is the number of inputs and  $P$  the number of outputs.

### 1.2.5 Nonlinear Wave Digital Filters

Wave digital filters are another approach to elimination or reduction of delay free loops. Wave digital filters were extensively studied in many publications. The detailed description and overview, including many practical examples, can be found in [36]. However, linear wave digital filters were the main interest there. Later publications discussed integration of nonlinear wave digital components to solve nonlinear problems. The extension of linear wave digital filters with a nonlinear resistance component for simulation of Chua's circuit was used in [37]. The nonlinear capacitor and the inductor model were introduced in [38] and later in [39]. An important step came with publication [30] where nonlinear wave digital filters were used for the real-time simulation of a tube single stage amplifier. This work was further extended in [17] and real-time simulation of a single ended tube power amplifier using wave digital filters was described in [31]. Finally, the overview of use of wave digital filters for the virtual analog models was given in [14].

The basic principle of wave digital filters is substitution of classic port variables – Kirchhoff variables (voltage  $v$  and current  $i$ ) by wave variables – incident and reflected waves  $A$  and  $B$  and a port resistance  $R$ , which is however not related with the physical resistance, all representing one port element (e.g. resistor). The transformation of the Kirchhoff variables to wave variables is given by

$$\begin{bmatrix} A \\ B \end{bmatrix} = \begin{bmatrix} 1 + R \\ 1 - R \end{bmatrix} \begin{bmatrix} v \\ i \end{bmatrix} \quad (1.34)$$

and the backward transformation

$$\begin{bmatrix} v \\ i \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 1 & 1 \\ \frac{1}{R} & -\frac{1}{R} \end{bmatrix} \begin{bmatrix} A \\ B \end{bmatrix}. \quad (1.35)$$

Then all circuit components are replaced by wave digital filters using the transformation (1.34). Reactive components, such as capacitors and inductors, further require component-wise discretization that makes use of the Bilinear transform. A library of transformed components can be found in literature [30, 31, 40] and others and therefore, it will not be further described here.

Once all components of the circuit have been transformed, they must be connected to form a wave digital filter structure. This process can lead to generation of delay free loops if the port resistances are unmatched because of instantaneous reflections. To cope with this problem, special adapters which allow matching of port resistances were introduced. Generally, it is possible to use  $N$ -port adapters, however it is usually sufficient to use only 3-port adapters connecting two ports and the rest of the system. Because connections in the electronic circuit can be either in series or parallel, both types of three port adapters were developed. Each adapter contains at



most one special port denoted as Reflection Free Port (RFP) whose impedance is matched to the resulting (series or parallel) impedance of all other ports connected to the adapter. The advantage of this port is that there is no instantaneous reflection at this port which could cause the delay free loop. Therefore, several parts of the circuit are connected using these RFP ports, forming a tree structure of port adapters with one reflection free port placed in the root of the tree. This tree can be found under the term Binary Connection Tree (BCT).

As it can be seen from the previous text, these adapters are essence of wave digital filter structures. The series  $N$ -port adapter can be described by scattering port relations given by scattering parameter

$$\gamma_v = \frac{2R_v}{R_1 + R_2 + \dots R_n} \quad (1.36)$$

for  $v$ -th port, the series junction wave variable

$$a_s = a_1 + a_2 + \dots a_n \quad (1.37)$$

and reflected waves for each port

$$b_v = a_v + \gamma_v a_s. \quad (1.38)$$

Similarly, the parallel adapter is given by

$$\gamma_v = \frac{2G_v}{G_1 + G_2 + \dots G_n} \quad (1.39)$$

for  $v$ -th port and where the symbol  $G$  denotes the conductance of the port, the parallel junction wave variable

$$a_p = \gamma_1 a_1 + \gamma_2 a_2 + \dots \gamma_n a_n \quad (1.40)$$

and reflected waves for each port

$$b_v = a_v + a_p. \quad (1.41)$$

When forming the Wave Digital Filter (WDF) tree manually, one must select the root of the tree and subsequently connect all circuit components using parallel and series adapters. The runtime equation can be formed using equations (1.36, 1.37, 1.38, 1.39, 1.40, 1.41) by substituting reflected waves into incident waves of connected blocks. The recomputation of incident wave variables starts from the most bottom leaves of the BCT up to the root and then all reflected waves are recomputed back to the bottom leaves of the BCT. As it has been already mentioned in [22], naming of variables in terms of the incident variable  $a$  and  $b$  is unwieldy and therefore it is more practical to label the wave going up to root by  $u$  wave and the wave going

down the tree as  $d$ . Using this notation, it is possible to rewrite 3-port adapters into a more convenient form

$$u = -(u_1 + u_2) \quad (1.42)$$

$$d_1 = u_1 - \gamma_1 (d - u) \quad (1.43)$$

$$d_2 = u_2 - \gamma_2 (d - u) \quad (1.44)$$

for series adaptor and

$$u = \gamma_1 u_1 + \gamma_2 u_2 \quad (1.45)$$

$$d_1 = u - d - u_1 \quad (1.46)$$

$$d_2 = u - d - u_2 \quad (1.47)$$

where  $u_1$  and  $u_2$  are up-going waves from connected ports to the adapter and  $d_1$ ,  $d_2$  are down-going waves. Variables  $u$  and  $d$  are up-going and down-going adapted waves to the next adapter and the root of the BCT. The recomputation starts with substituting inputs and delayed reflected waves of accumulating elements into the up-going wave and afterwards, all accumulating elements are updated by down-going wave and the state is again stored by the delayed reflected wave. A big advantage of using these adapters is the low computational complexity. According to [40], the  $N$ -port adapter requires  $N - 1$  multiplications and  $2N - 2$  additions and  $N$  element circuits  $N - 2$  adapters leading to the overall complexity  $2N - 4$  multiplications and  $4N - 8$  additions. This means that the WDF offers the linear computational complexity in contrast to the general scattering matrices providing quadratic computational complexity.

The manual derivation of runtime equations is rather demanding. Therefore, literature [41, 42] deals with a systematic way of derivation of the WDF structure. Literature [42] also deals with an implementation of wave digital filters in the BlockCompiler software tool [43, 44], which was also used for the real-time simulation of tube amplifiers in papers [30, 17, 31]. Using this tool, it is possible to obtain a very efficient implementation of simulation the of the given circuit, even if it is nonlinear.

Nonlinear wave digital filters are the extension of linear digital filters by nonlinear components. In case of audio effect circuit it would be nonlinear models of tubes, diodes and transistors that are modeled as a nonlinear resistance in WDF structure. The port nonlinear resistance is a reflection function of the incoming wave. Considering incorporation of the nonlinear element in the BCT requires the nonlinear component to be matched with the adapter to which it is connected. However, the nonlinear port resistance is the function of the incoming wave and thus, it requires to recompute scattering parameters of all adapters in the BCT to

match port impedances every time when the resistance changes, which degrades the overall computational efficiency of the algorithm. Therefore, when an efficient algorithm is desired, the nonlinear element must be connected to the reflection free port to prevent the propagation of the resistance change to all other elements. It means, according to the connection rules, that the nonlinear element must be placed as the root element. This fact implies that it is possible to make the efficient algorithm only with one nonlinear function, which is main drawback of this method. Although modifications of WDF based on a multi-port vector nonlinear element were introduced ([45]), multiple nonlinear circuits are solved with an unit delay of nonlinear functions [30, 46, 14]. However, this simplification is essentially the same as the use of explicit ODE and it can lead to non-stability of the solution under certain circumstances unless a sufficiently high sampling frequency is used.

### 1.2.6 Volterra Series

A nonlinear convolution is one of other methods successfully applied for the simulation of nonlinear audio effects. Although this approach is quite old and well known, it has been exploited recently because more powerful processors are available. Nevertheless, in contrast with the previously mentioned algorithms, it is the black-box approach, which needs techniques for the nonlinear system identification.

The base of this approach is the Volterra series, which is the extension of the classical linear system theory into nonlinear systems. The necessary theory can be found e.g. in [47]. The discrete time Volterra series is given by

$$\begin{aligned}
y[n] = & \sum_{m_1=0}^M h_1[m_1]x[n - m_1] + \\
& + \sum_{m_1=0}^M \sum_{m_2=0}^M h_2[m_1, m_2]x[n - m_1]x[n - m_2] + \dots \\
& + \sum_{m_1=0}^M \dots \sum_{m_N=0}^M h_N[m_1, \dots, m_N]x[n - m_1] \dots x[n - m_N]
\end{aligned} \tag{1.48}$$

with  $M \rightarrow \infty$  and where  $h_1, h_2$  to  $h_N$  are up to  $N$ -dimensional Volterra kernels. This equation is capable to describe a nonlinear system with memory. Implementation of the algorithm is very straightforward according to (1.48) but on the other hand, obtaining of Volterra kernels is a very difficult task. The Volterra kernels can be obtained using cross-correlations of the pseudo-random input signal and output signal as was suggested in [14], by nonlinear adaptive filters [48] or even by derivation from circuit equations [49]. In all cases, a strongly linear system require the Volterra series of the high order which leads to rapid increasing of computational cost and also the amount of coefficients of  $N$ -dimensional Volterra kernels.

However, it is possible to reduce dimension of the Volterra kernels by their factorization [14] given by

$$h_i(n_1, n_2, \dots, n_i) = \prod_i h_i^f(n) \quad (1.49)$$

where the original  $N$ -dimensional Volterra kernel is expressed as a product of factorized one dimensional kernels  $h_i^f(n)$  and the Volterra series can be rewritten into

$$\begin{aligned} y[n] &= \sum_{m=0}^M h_1^f[m]x[n-m] + \\ &+ \left( \sum_{m=0}^M h_1^f[m]x[n-m] \right)^2 + \dots + \left( \sum_{m=0}^M h_N^f[m]x[n-m] \right)^N = \\ &= \sum_i^N \left( \sum_{m=0}^M h_i^f[m]x[n-m] \right)^i \end{aligned} \quad (1.50)$$

which can be decomposed into the linear dynamic and nonlinear memoryless parts (see Figure 1.1) and it can be treated as the Wiener nonlinear model.

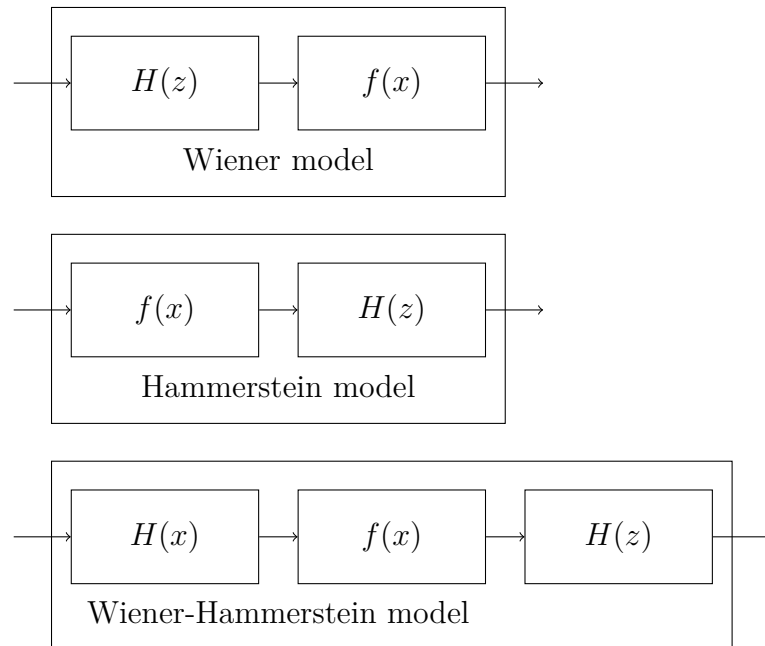


Figure 1.1: Block diagrams of Wiener (top), Hammerstein (middle) and Wiener-Hammerstein nonlinear model.

This model as well as the identification of filter coefficients for the simulation of nonlinear audio amplifier was discussed in [50]. This simplification allows significant reduction of computational cost while the the accuracy of simulation is sufficient for certain type of circuits. The further advantage is that this implementation enables

to avoid the aliasing distortion by modifying linear filters by incorporating low-pass filters. As can be seen from Figure 1.1, there are other simplified models of Volterra series, namely the Hammerstein model and the Wiener-Hammerstein model. The Hammerstein model is given by

$$\begin{aligned}
y[n] &= \sum_{m=0}^M h_1^f[m]x[n-m] + \\
&+ \sum_{m=0}^M h_1^f[m]x^2[n-m] + \cdots + \sum_{m=0}^M h_N^f[m]x^N[n-m] = \quad (1.51) \\
&= \sum_i^N \left( \sum_{m=0}^M h_i^f[m]x^i[n-m] \right)
\end{aligned}$$

and this algorithm is called the nonlinear convolution in literature [51]. It was also used in paper [52] where the audio limiter effect was simulated using this approach and in later papers [53, 54] where the overdrive audio effect was simulated. In contrary to the former paper, the memoryless nonlinearity was implemented by Chebyshev polynomials. Results showed that it is possible to obtain good accuracy with the Volterra model with order 9. However neither real-time conditions nor the computational complexity were discussed.

The proper determination of system coefficients is essential for the sufficient accuracy of simulation. Plenty of papers in different fields of research deal with an identification of Hammerstein model parameters. The method convenient for the identification of audio systems was presented in Farina's work [51]. It is based on the swept-sine signal excitation of the measured system. Subsequently, deconvolution of the obtained signal response with the inverse filter signal (the time reversal excitation signal with the slope of 6db/oct) is computed to get the nonlinear impulse response which contains the impulse response of linearized system and then higher order impulse responses. Finally, it is possible to extract Hammerstein filter kernels using some algebra described in the paper. This method was further extended by the work [52, 53, 54] resulting in the synchronized swept-sine method. And even when one decides to use another approach for the simulation of the audio effect, this approach still remains an useful tool for the distortion analysis. To sum up this approach, the published results [52, 53, 54] showed that it is possible to obtain the accurate simulation of audio effects. However, all the simulated effects contained one nonlinearity surrounded by linear circuit components, which is in correspondence with the Wiener-Hammerstein model of the nonlinear system. The system containing more nonlinear blocks connected with linear dynamic parts would probably require use of the Volterra series in the full form.

### 1.3 Recent Advances

Although the real-time simulation of analog audio effects is very specific area of research, there have recently been many publications covering all types of algorithms mentioned in previous chapters and also attempting new ones. A good overview can be found in [55]. The method based on time-invariant filter was published by me in [3]. The simulated circuit, the diode limiter circuit in this case, is described as the small-signal model using the transfer function where the nonlinear component was substituted by the resistance in its operating point. In case of small-signals, it is possible to compute the output signal directly from the difference equation. However, the operating point changes with large-signals consequently affecting the value of the resistance and thus, coefficients of the filter change according to the signal value on the nonlinear element. Two versions of the algorithm were proposed. The first one uses the signal value from the previous sampling period and requires a very high sampling frequency to operate properly. The other one uses the signal value on the nonlinear element from the current sampling period which leads to generation of the delay free loop and thus, the numerical algorithm must be used. This approach evinces similar properties as direct solving of ODEs and therefore it is not very convenient for the simulation of more complex systems.

The static waveshaping method was used in [56] for the simulation of a diode ring modulator circuit and in [57] for the simulation of bucket-brigade device circuits. However both papers use techniques which were already discussed in chapter 1.2.3. An improvement of the wave-shaping method was the goal of work presented in [58]. This method makes use of the standard wave-shaping method implemented as the sum of Chebyshev polynomials and it also discusses usage of the phase-shaping of the input signal, which is also applicable as the nonlinear distortion effect [59]. The main idea was to use a quadrature waveshaper to incorporate the phase information, which is not possible with the regular waveshaper. Coefficients required by the quadrature waveshapers are determined from the measurement of the simulated distortion effect.

Another black-box approach based on the Principal Component Analysis (PCA) is published in [60]. This method makes use of the swept-sine technique for the measurement of the distortion effect as was described in [52, 53, 54] but in contrast with these papers, the simplification of measured results through the PCA is used, resulting in a lower order system than if it would have to be used in case of direct application of the Hammerstein model. This is possible due to the compression of the measured information. The system has the similar structure to the Hammerstein model consisting of  $M$  branches, each with the static waveshaper and the linear filter given by its impulse response. The  $N$  original Chebyshev polynomials were replaced by  $M$  components computed from the PCA and also  $N$  original higher order

impulse responses were transformed into  $M$  compressed impulse responses. If the compression is possible, then  $M < N$ . This approach was tested in the simulation of the Tube Screamer overdrive guitar effect pedal and the original system of order 27 was reduced to the new system of order 8 and overall computational cost decreased by 66 %.

There has also been a progress in the field of the WDF approach. The implementation of the WDF structure in the Csound audio programming language was shown in [61]. The implementation was verified in the simulation of the common cathode triode amplifier with both nonlinear grid and plate currents and it is workable in real-time. The detailed overview on the virtual analog modeling was given in literature [62], where wide range of WDF blocks were reviewed. Further, the problem of the multiple nonlinearity in the WDF structure is discussed on the example of the common emitter transistor amplifier. The standard approach of connecting one port elements with 3-port adapters leads to the non-computable loop. However, it was shown that it is possible to solve this circuit efficiently by using a multiple port nonlinear block connected in the root of the BCT. The options of usage of WDFs are shown also on other examples. The WDF based simulation of the tube power amplifier was published in paper [46]. This simulation also incorporates the nonlinear model of the output transformer based on the gyrator-capacitor model. This model consists of two nonlinear functions which were however delayed by one sample, which can be considered as a step back, when comparing to literature [62] where multiple nonlinearities are discussed. However this paper is important because of the design of the nonlinear transformer model for real-time processing.

### 1.3.1 Advances in State-Space Modeling

A big step ahead has been done in the area of state-space modeling algorithms. A modified approach of the state-space modeling was introduced by Cohen in his work [63, 64]. The original linear state-space model is given by

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{Ax} + \mathbf{Bu} \\ \mathbf{y} &= \mathbf{Cx} + \mathbf{Du}\end{aligned}\tag{1.52}$$

where  $\mathbf{x}$  is the state vector,  $\mathbf{y}$  is the output vector and  $\mathbf{u}$  is the input vector which is extended by the static nonlinear vector  $\mathbf{w}$  forming the new state-space equations

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{f}(\mathbf{x}, \mathbf{u}, \mathbf{w}) \\ \mathbf{0} &= \mathbf{g}(\mathbf{x}, \mathbf{u}, \mathbf{w}) \\ \mathbf{y} &= \mathbf{h}(\mathbf{x}, \mathbf{u}, \mathbf{w})\end{aligned}\tag{1.53}$$

where functions  $\mathbf{f}$ ,  $\mathbf{h}$  would be for the linear case  $\mathbf{f}(\mathbf{x}, \mathbf{u}, \mathbf{w}) = \mathbf{Ax} + \mathbf{Bu}$ ,  $\mathbf{h}(\mathbf{x}, \mathbf{u}, \mathbf{w}) = \mathbf{Cx} + \mathbf{Du}$ ,  $\mathbf{g}$  unused and  $\mathbf{w} = \mathbf{0}$ . In case of nonlinear systems, they must be determined

manually according to the given circuit. To implement equation (1.53), it must be discretized first. The trapezoidal rule of the first equation in the system of equations (1.53) leads to the implicit state update function

$$\mathbf{x}[n+1] = \mathbf{x}[n] + \frac{T}{2} f(\mathbf{x}[n+1], \mathbf{u}[n+1], \mathbf{w}[n+1]) + \frac{T}{2} f(\mathbf{x}[n], \mathbf{u}[n], \mathbf{w}[n]) \quad (1.54)$$

with the sampling period  $T$ . This implicit equation requires the numerical algorithm to solve and for the use of the numerical algorithm can be rewritten to  $\mathbf{f}(\mathbf{Z}) = \mathbf{0}$  where  $\mathbf{Z} = \begin{bmatrix} \mathbf{x} & \mathbf{w} \end{bmatrix}^T$ . Although this state-space representation is more general, the dimension of the vector of unknowns  $\mathbf{Z}$  to be solved numerically is higher than with the DK-method. It was also found in later literature [65] that the numerical scheme has problems with the stiffness and it can be unstable. Therefore, the BE discretization was used to get the state update function

$$\mathbf{x}[n+1] = \mathbf{x}[n] + T f(\mathbf{x}[n+1], \mathbf{u}[n+1], \mathbf{w}[n+1]). \quad (1.55)$$

Using this technique, Cohen was able to numerically simulate the guitar preamp with more tubes connected in series in real-time. The simulation requires the sampling frequency at least 96 kHz, but the higher sampling frequency should be used to cope with the aliasing distortion anyway.

A deep review of the DK-method was done by Holters and Zölzer in [66]. The audio circuit is described using incidence matrices which specify to which nodes are circuit components connected. Since there have been more types of the circuit components, resistors are specified in the incidence matrix  $\mathbf{N}_r$ , accumulating components in  $\mathbf{N}_x$  matrix, nonlinear components in the matrix  $\mathbf{N}_n$ , inputs of the circuit, such as the input signal and the power supply voltage, are specified in  $\mathbf{N}_u$  and finally, outputs of the circuit are given by  $\mathbf{N}_o$  matrix. The circuit equation can be expressed as

$$\mathbf{S} \begin{pmatrix} \mathbf{v} \\ \mathbf{i}_s \end{pmatrix} = \begin{pmatrix} \mathbf{N}_x^T \\ \mathbf{0} \end{pmatrix} \mathbf{x} + \begin{pmatrix} \mathbf{0} \\ \mathbf{I} \end{pmatrix} \mathbf{u} + \begin{pmatrix} \mathbf{N}_n^T \\ \mathbf{0} \end{pmatrix} \mathbf{i}_n \quad (1.56)$$

which has got the same formal structure as the MNA circuit equation given by (1.4) and where  $\mathbf{i}_n$  is the vector of nonlinear currents,  $\mathbf{I}$  is the identity matrix,  $\mathbf{u}$  is the vector of input voltages,  $\mathbf{x}$  is the vector of states,  $\mathbf{i}_s$  is the vector of unknown currents of ideal voltage sources,  $\mathbf{v}$  is the vector of unknown voltages and  $\mathbf{S}$  is the conductance matrix given by

$$\mathbf{S} = \begin{pmatrix} \mathbf{N}_r^T \mathbf{G}_r \mathbf{N}_r + \mathbf{N}_x^T \mathbf{G}_x \mathbf{N}_x & \mathbf{N}_u^T \\ \mathbf{N}_u & \mathbf{0} \end{pmatrix}. \quad (1.57)$$

The matrix  $\mathbf{G}_r$  is the diagonal resistor conductance matrix and  $\mathbf{G}_x$  is the diagonal capacitor resp. inductor conductance matrix. The voltages on accumulating



components  $\mathbf{v}_x$  can be obtained from

$$\mathbf{v}_x = \begin{pmatrix} \mathbf{N}_x & \mathbf{0} \end{pmatrix} \mathbf{S}^{-1} \left( \begin{pmatrix} \mathbf{N}_x^T \\ \mathbf{0} \end{pmatrix} \mathbf{x} + \begin{pmatrix} \mathbf{0} \\ \mathbf{I} \end{pmatrix} \mathbf{u} + \begin{pmatrix} \mathbf{N}_n^T \\ \mathbf{0} \end{pmatrix} \mathbf{i}_n \right) \quad (1.58)$$

and similarly, voltages  $\mathbf{v}_n$  and  $\mathbf{v}_o$  can be computed when incident matrix  $\mathbf{N}_n$  resp.  $\mathbf{N}_o$  are used instead of  $\mathbf{N}_x$ . Using the voltages  $\mathbf{v}_x$ ,  $\mathbf{v}_n$ ,  $\mathbf{v}_o$ , discretized accumulating component models (refer chapter 1.4 for details) and DK-method equations, it is possible to derive for the TR discretization following equations

$$\mathbf{x}[n] = \mathbf{A}\mathbf{x}[n-1] + \mathbf{B}\mathbf{u}[n] + \mathbf{C}\mathbf{i}_n(\mathbf{v}[n]), \quad (1.59)$$

$$\mathbf{y}[n] = \mathbf{D}\mathbf{x}[n-1] + \mathbf{E}\mathbf{u}[n] + \mathbf{F}\mathbf{i}_n(\mathbf{v}[n]), \quad (1.60)$$

$$\mathbf{v}[n] = \mathbf{G}\mathbf{x}[n-1] + \mathbf{H}\mathbf{u}[n] + \mathbf{K}\mathbf{i}_n(\mathbf{v}[n]) \quad (1.61)$$

with matrices

$$\mathbf{A} = 2\mathbf{Z}\mathbf{G}_x \begin{pmatrix} \mathbf{N}_x & \mathbf{0} \end{pmatrix} \mathbf{S}^{-1} \begin{pmatrix} \mathbf{N}_x & \mathbf{0} \end{pmatrix}^T - \mathbf{Z}, \quad (1.62)$$

$$\mathbf{B} = 2\mathbf{Z}\mathbf{G}_x \begin{pmatrix} \mathbf{N}_x & \mathbf{0} \end{pmatrix} \mathbf{S}^{-1} \begin{pmatrix} \mathbf{0} & \mathbf{I} \end{pmatrix}^T, \quad (1.63)$$

$$\mathbf{C} = 2\mathbf{Z}\mathbf{G}_x \begin{pmatrix} \mathbf{N}_x & \mathbf{0} \end{pmatrix} \mathbf{S}^{-1} \begin{pmatrix} \mathbf{N}_n & \mathbf{0} \end{pmatrix}^T, \quad (1.64)$$

$$\mathbf{D} = \begin{pmatrix} \mathbf{N}_o & \mathbf{0} \end{pmatrix} \mathbf{S}^{-1} \begin{pmatrix} \mathbf{N}_x & \mathbf{0} \end{pmatrix}^T, \quad (1.65)$$

$$\mathbf{E} = \begin{pmatrix} \mathbf{N}_o & \mathbf{0} \end{pmatrix} \mathbf{S}^{-1} \begin{pmatrix} \mathbf{0} & \mathbf{I} \end{pmatrix}^T, \quad (1.66)$$

$$\mathbf{F} = \begin{pmatrix} \mathbf{N}_o & \mathbf{0} \end{pmatrix} \mathbf{S}^{-1} \begin{pmatrix} \mathbf{N}_n & \mathbf{0} \end{pmatrix}^T, \quad (1.67)$$

$$\mathbf{G} = \begin{pmatrix} \mathbf{N}_n & \mathbf{0} \end{pmatrix} \mathbf{S}^{-1} \begin{pmatrix} \mathbf{N}_x & \mathbf{0} \end{pmatrix}^T, \quad (1.68)$$

$$\mathbf{H} = \begin{pmatrix} \mathbf{N}_n & \mathbf{0} \end{pmatrix} \mathbf{S}^{-1} \begin{pmatrix} \mathbf{0} & \mathbf{I} \end{pmatrix}^T, \quad (1.69)$$

$$\mathbf{K} = \begin{pmatrix} \mathbf{N}_n & \mathbf{0} \end{pmatrix} \mathbf{S}^{-1} \begin{pmatrix} \mathbf{N}_n & \mathbf{0} \end{pmatrix}^T \quad (1.70)$$

where  $\mathbf{Z}$  is the diagonal matrix with 1 for the capacitor and  $-1$  for the inductor with size equal to the size of  $\mathbf{G}_x$ . Note that the notation of the matrices slightly differs from the original DK method.

### 1.3.2 State-Space Approach for Parametric Circuits

State-space approaches discussed in previous chapters were focused on circuits with constant circuit parameters. However, most of audio effect circuits are parametric circuits. The application of the standard DK-method on simulation of the parametric circuit was discussed in [67]. The change of parameter in the standard DK-method requires the recomputation of the whole circuit matrix which is often huge even for

simple circuits. Therefore, the derivation of the DK-method matrices from continuous state-space description was introduced and tested on two types of audio circuits – the linear tone stack circuit and the distortion stage of the Marshall JCM 900 amplifier. Firstly, continuous state-space matrices are derived symbolically from circuit equations where states are voltages on capacitors. Then the discretization follows. For the TR discretization, it is possible to derive the DK-method matrices (denoted with the line over the matrix symbol)

$$\bar{\mathbf{A}} = \left( \frac{2}{T} \mathbf{I} + \mathbf{A} \right) \left( \frac{2}{T} \mathbf{I} - \mathbf{A} \right)^{-1} \quad (1.71)$$

$$\bar{\mathbf{B}} = 2 \left( \frac{2}{T} \mathbf{I} - \mathbf{A} \right)^{-1} \mathbf{B} \quad (1.72)$$

$$\bar{\mathbf{C}} = 2 \left( \frac{2}{T} \mathbf{I} - \mathbf{A} \right)^{-1} \mathbf{C} \quad (1.73)$$

$$\bar{\mathbf{D}} = \frac{2}{T} \mathbf{D} \left( \frac{2}{T} \mathbf{I} - \mathbf{A} \right)^{-1} \quad (1.74)$$

$$\bar{\mathbf{E}} = \mathbf{E} + \mathbf{D} \left( \frac{2}{T} \mathbf{I} - \mathbf{A} \right)^{-1} \mathbf{B} \quad (1.75)$$

$$\bar{\mathbf{F}} = \mathbf{F} + \mathbf{D} \left( \frac{2}{T} \mathbf{I} - \mathbf{A} \right)^{-1} \mathbf{C} \quad (1.76)$$

$$\bar{\mathbf{G}} = \frac{2}{T} \mathbf{G} \left( \frac{2}{T} \mathbf{I} - \mathbf{A} \right)^{-1} \quad (1.77)$$

$$\bar{\mathbf{H}} = \mathbf{H} \mathbf{G} \left( \frac{2}{T} \mathbf{I} - \mathbf{A} \right)^{-1} \mathbf{B} \quad (1.78)$$

$$\bar{\mathbf{K}} = \mathbf{K} \mathbf{G} \left( \frac{2}{T} \mathbf{I} - \mathbf{A} \right)^{-1} \mathbf{C}. \quad (1.79)$$

Compared to the automated nodal DK-method presented by Yeh, this approach offers lower computational cost for the recomputation of the DK-method matrices when there is a change of circuit parameters. However, the main drawback is symbolical expression of state-space matrices.

An efficient handling of parameter changes was also considered in the reviewed DK-method by Holters and Zölzer [66] where the DK-method with incidence component matrices was used for the simulation of a guitar wah pedal, which is an example of quite complex and parametric audio effect circuit. The standard approach of the DK-method matrices recomputation would also lead to computationally demanding process. However, it is possible to separate constant and parametric part of the circuit which leads to the modification of conductance matrix  $\mathbf{S}$  into

$$\begin{aligned} \mathbf{S} &= \begin{pmatrix} \mathbf{N}_r^T \mathbf{G}_r \mathbf{N}_r + \mathbf{N}_v^T \mathbf{R}_v^{-1} \mathbf{N}_v + \mathbf{N}_x^T \mathbf{G}_x \mathbf{N}_x & \mathbf{N}_u^T \\ \mathbf{N}_u & \mathbf{0} \end{pmatrix} = \\ &= \mathbf{S}_0 + \begin{pmatrix} \mathbf{N}_v & \mathbf{0} \end{pmatrix}^T \mathbf{R}_v^{-1} \begin{pmatrix} \mathbf{N}_v & \mathbf{0} \end{pmatrix} \end{aligned} \quad (1.80)$$

where  $\mathbf{N}_v$  is the incidence matrix of variable resistors (potentiometers) and  $\mathbf{R}_v$  diagonal matrix of corresponding resistances and  $\mathbf{S}_0$  the constant conductance matrix given exactly according to (1.57). The conductance matrix inversion than can be expressed as

$$\mathbf{S}^{-1} = \mathbf{S}_0^{-1} - \mathbf{S}_0^{-1} \begin{pmatrix} \mathbf{N}_v & \mathbf{0} \end{pmatrix}^T (\mathbf{R}_v^{-1} + \mathbf{Q}) \begin{pmatrix} \mathbf{N}_v & \mathbf{0} \end{pmatrix} \mathbf{S}_0^{-1} \quad (1.81)$$

with

$$\mathbf{Q} = \begin{pmatrix} \mathbf{N}_v & \mathbf{0} \end{pmatrix} \mathbf{S}_0^{-1} \begin{pmatrix} \mathbf{N}_v & \mathbf{0} \end{pmatrix}^T. \quad (1.82)$$

Final matrices can be obtained from

$$\mathbf{A} = \mathbf{A}_0 - 2\mathbf{Z}\mathbf{G}_x\mathbf{U}_x (\mathbf{R}_v + \mathbf{Q})^{-1} \mathbf{U}_x^T, \quad (1.83)$$

$$\mathbf{B} = \mathbf{B}_0 - 2\mathbf{Z}\mathbf{G}_x\mathbf{U}_x (\mathbf{R}_v + \mathbf{Q})^{-1} \mathbf{U}_u^T, \quad (1.84)$$

$$\mathbf{C} = \mathbf{C}_0 - 2\mathbf{Z}\mathbf{G}_x\mathbf{U}_x (\mathbf{R}_v + \mathbf{Q})^{-1} \mathbf{U}_n^T, \quad (1.85)$$

$$\mathbf{D} = \mathbf{D}_0 - \mathbf{U}_o (\mathbf{R}_v + \mathbf{Q})^{-1} \mathbf{U}_x^T, \quad (1.86)$$

$$\mathbf{E} = \mathbf{E}_0 - \mathbf{U}_o (\mathbf{R}_v + \mathbf{Q})^{-1} \mathbf{U}_u^T, \quad (1.87)$$

$$\mathbf{F} = \mathbf{F}_0 - \mathbf{U}_o (\mathbf{R}_v + \mathbf{Q})^{-1} \mathbf{U}_n^T, \quad (1.88)$$

$$\mathbf{G} = \mathbf{G}_0 - \mathbf{U}_n (\mathbf{R}_v + \mathbf{Q})^{-1} \mathbf{U}_x^T, \quad (1.89)$$

$$\mathbf{H} = \mathbf{H}_0 - \mathbf{U}_n (\mathbf{R}_v + \mathbf{Q})^{-1} \mathbf{U}_u^T, \quad (1.90)$$

$$\mathbf{K} = \mathbf{K}_0 - \mathbf{U}_n (\mathbf{R}_v + \mathbf{Q})^{-1} \mathbf{U}_n^T, \quad (1.91)$$

where  $\mathbf{A}_0$ ,  $\mathbf{B}_0$ ,  $\mathbf{C}_0$ ,  $\mathbf{D}_0$ ,  $\mathbf{E}_0$ ,  $\mathbf{F}_0$ ,  $\mathbf{G}_0$ ,  $\mathbf{H}_0$  and  $\mathbf{K}_0$  are matrices for constant parameters computed according to (1.62 - 1.70) and

$$\mathbf{U}_x = \begin{pmatrix} \mathbf{N}_x & \mathbf{0} \end{pmatrix} \mathbf{S}_0^{-1} \begin{pmatrix} \mathbf{N}_v & \mathbf{0} \end{pmatrix}^T \quad (1.92)$$

$$\mathbf{U}_o = \begin{pmatrix} \mathbf{N}_o & \mathbf{0} \end{pmatrix} \mathbf{S}_0^{-1} \begin{pmatrix} \mathbf{N}_v & \mathbf{0} \end{pmatrix}^T \quad (1.93)$$

$$\mathbf{U}_n = \begin{pmatrix} \mathbf{N}_n & \mathbf{0} \end{pmatrix} \mathbf{S}_0^{-1} \begin{pmatrix} \mathbf{N}_v & \mathbf{0} \end{pmatrix}^T \quad (1.94)$$

$$\mathbf{U}_u = \begin{pmatrix} \mathbf{0} & \mathbf{I} \end{pmatrix} \mathbf{S}_0^{-1} \begin{pmatrix} \mathbf{N}_v & \mathbf{0} \end{pmatrix}^T. \quad (1.95)$$

Although this equation might seem very complex, the significant computational cost reduction can be achieved because the size of matrix  $\mathbf{Q}$  is usually much lower than the size of complex system.

## 1.4 Basic Circuit Component Models for Real-time Audio Effect Simulation

The choice of a proper model of circuit components is one of the most important aspects when simulating the analog audio effect. A wide range of different circuit

components with different complexity has been designed which allow to achieve very accurate simulation results. However, this complexity is in the contrast with requirements of audio real-time processing. Fortunately, it is possible to neglect several aspects of the complex model – foremost parasitic capacitances of the models because of limited frequency bandwidth. This enables to design models that are frequency independent and it is possible to describe them using the nonlinear static mapping of the input to the output variable. When talking about component models, one usually think only of nonlinear component models. However, as it was mentioned in previous chapters, many of simulation algorithms require the component-wise discretization of energy-storing components which are defined by differential equations. Therefore, in this chapter, the local discretization of linear components will be described and further some nonlinear models convenient for use in real-time digital audio signal processing will be discussed.

### 1.4.1 Discretized Models of Capacitor and Inductor

The discretization of energy-storing components is often made by using companion circuits consisting of the equivalent conductance and the source holding the state of the component as was shown in [68] and was also derived in [22]. As a result of application of TR on the capacitor current equation

$$i_C = C \frac{dv}{dt}, \quad (1.96)$$

the capacitor companion model is

$$i[n] = \frac{2C}{T}v + i_C[n-1] \quad (1.97)$$

where

$$i_C[n] = -\left(\frac{2C}{T}v + i[n]\right). \quad (1.98)$$

Similarly for the inductor, the companion circuit is

$$v[n] = \frac{2}{T}\Phi[n] + v_L[n-1] \quad (1.99)$$

where

$$v_L[n] = -\left(\frac{2}{T}\Phi[n] + v[n]\right) \quad (1.100)$$

and  $\Phi[n]$  is flux given by  $Li[n]$  with the inductance  $L$ .

To unify both models, Holters and Zölzer have suggested following discretization scheme [66]

$$i[n] = G_x v[n] + x[n-1] \quad (1.101)$$

where  $i[n]$  is the current flowing through the capacitor or the inductor,  $G_x = \frac{2C}{T}$  for the capacitor or  $G_x = \frac{T}{2L}$  for the inductor,  $v$  is voltage on the element and  $x$  is a canonical state of the element. The state update is possible using

$$x[n] = Z (2G_x v[n] - x[n-1]) \quad (1.102)$$

where  $Z = 1$  for the capacitor and  $Z = -1$  for the inductor.

## 1.4.2 Diode Model

The simulation of circuits with the embedded diode was discussed in many papers, e.g. [22, 24, 32, 67] and all have shown that it is possible to use the simplified model of the diode given by Shockley equation

$$i = i_S \left( \exp \left( \frac{v}{n v_T} \right) - 1 \right) \quad (1.103)$$

where  $i_S$  is the saturation current,  $v_T$  the thermal voltage and  $n$  the emission coefficient. Since diodes have often been used in anti-parallel topology in audio effect circuits, the resulting current of anti-parallel connection can be expressed as

$$i = 2i_S \sinh \left( \frac{v}{n v_T} \right). \quad (1.104)$$

## 1.4.3 Transistor Model

Transistors, either bipolar junction or junction field effect transistors, are often used as signal buffers or as active elements in filters. A nonlinear DC model of the Bipolar Junction Transistor (BJT) can be modeled by well known Ebers-Moll equations

$$\begin{aligned} i_C &= i_{CC} - \frac{\beta_R + 1}{\beta_R} i_{CE} \\ i_E &= i_{CE} - \frac{\beta_F + 1}{\beta_F} i_{CC} \\ i_B &= \frac{1}{\beta_F} i_{CC} + \frac{1}{\beta_R} i_{CE} \end{aligned} \quad (1.105)$$

where currents  $i_{CC}$  and  $i_{CE}$  are given by Shockley equation (1.103) for voltages  $v_{BE}$  resp.  $v_{BC}$  and parameters  $\beta_F$  and  $\beta_R$  are forward and reverse current gain parameters. This model was successfully tested on simulation of the guitar wah pedal [66].

The Junction Field Effect Transistor (JFET) can be modeled e.g. using Shichman-Hodges model which is also used in the SPICE simulator [68]. The Shichman-Hodges model is defined

$$i_{DS} = \begin{cases} 0 & v_{GS} \leq v_{TO} \\ \beta (v_{GS} - v_{TO})^2 (1 + \lambda v_{DS}) & 0 < v_{GS} - v_{TO} \leq v_{DS} \\ \beta v_{DS} (2(v_{GS} - v_{TO}) - v_{DS}) (1 + \lambda v_{DS}) & 0 < v_{DS} < v_{GS} - v_{TO} \end{cases} \quad (1.106)$$

where  $i_{\text{DS}}$  is the drain-source current,  $v_{\text{GS}}$  is the gate-source voltage,  $v_{\text{DS}}$  is the drain-source voltage and  $v_{\text{TO}}$ ,  $\beta$ ,  $\lambda$  are the threshold voltages, the transconductance factor and the conductance factor in saturation respectively.

#### 1.4.4 Tube Models

Because there has been recently quite big attention paid for the simulation of circuits containing tubes, a wide range of tube models has been discussed in many papers [69, 70, 64, 65, 71, 72] and others. Models proposed by Koren [70] and its extended version with a grid current designed by Cohen [65] and the triode model proposed by Dempwolf [69] seem to be the most perspective. The Koren's triode model was used in many publications dealing with the simulation of triode amplifier e.g. [40, 22, 30, 64, 65] and its validity was proved by measurement in [65]. The model is given by the plate current function

$$i_{\text{p}} = \frac{E_1^{E_x}}{K_{\text{g1}}} (1 + \text{sgn}(E_1)) \quad (1.107)$$

where

$$E_1 = \frac{v_{\text{pk}}}{K_{\text{p}}} \log\left(1 + \exp\left(K_{\text{p}}\left(\frac{1}{\mu} + \frac{v_{\text{gk}}}{\sqrt{K_{\text{vb}} + v_{\text{pk}}^2}}\right)\right)\right) \quad (1.108)$$

with tube parameters  $\mu$ ,  $E_x$ ,  $K_{\text{p}}$  and  $K_{\text{vb}}$  and  $v_{\text{gk}}$  is the grid-to-cathode voltage and  $v_{\text{pk}}$  is the plate-to-cathode voltage. The grid current added by Cohen is

$$i_{\text{g}} = \begin{cases} 0 & v_{\text{gk}} < v_{\gamma} - K_{\text{n}} \\ av_{\text{gk}}^2 + bv_{\text{gk}} + c & v_{\gamma} - K_{\text{n}} < v_{\text{gk}} < v_{\gamma} + K_{\text{n}} \\ \frac{v_{\text{gk}} - v_{\gamma}}{R_{\text{gk}}} & v_{\text{gk}} > v_{\gamma} + K_{\text{n}} \end{cases} \quad (1.109)$$

where

$$\begin{aligned} a &= \frac{1}{4K_{\text{n}}R_{\text{gk}}} \\ b &= \frac{K_{\text{n}} - v_{\gamma}}{2K_{\text{n}}R_{\text{gk}}} \\ c &= -a(v_{\gamma} - K_{\text{n}})^2 - b(v_{\gamma} - K_{\text{n}}) \end{aligned} \quad (1.110)$$

for tube parameters  $K_{\text{n}}$ ,  $v_{\gamma}$  and  $R_{\text{gk}}$ .

The Koren's model is also defined for a pentode tube given by the plate current

$$i_{\text{p}} = \frac{E_1^{E_x}}{K_{\text{g1k}}} (1 + \text{sgn}(E_1)) \arctan\left(\frac{v_{\text{pk}}}{K_{\text{vb}}}\right), \quad (1.111)$$

where

$$E_1 = \frac{v_{\text{g2k}}}{K_{\text{p}}} \log\left(1 + \exp\left(K_{\text{p}}\left(\frac{1}{\mu} + \frac{v_{\text{g1k}}}{v_{\text{g2k}}}\right)\right)\right) \quad (1.112)$$

and the pentode screen current is given by

$$i_s = \frac{\exp(E_x \log(\frac{v_{g2k}}{\mu} + v_{g1k}))}{K_{g2}} \quad (1.113)$$

with tube parameters  $\mu$ ,  $E_x$ ,  $K_{g1}$ ,  $K_{g2}$ ,  $K_p$  and  $K_{vb}$ . The Cohen grid model can be used for modeling the grid current of the pentode.

The Dempwolf's triode model [69], which is physically motivated, is given by

$$\begin{aligned} i_g &= -G_g \cdot \left( \log \left( 1 + \exp(C_g v_{gk}) \right) \cdot \frac{1}{C_g} \right)^{\xi_g} \\ i_p &= -G_k \cdot \left( \log \left( 1 + \exp \left( C_k \left( \frac{v_{pk}}{\mu} + v_{gk} \right) \right) \right) \cdot \frac{1}{C_k} \right)^{\xi_k} - i_g \end{aligned}$$

for tube parameters  $G_g$ ,  $C_g$ ,  $\xi_g$ ,  $G_p$ ,  $C_p$ ,  $\xi_p$  and  $\mu$  and was also verified by measurements. Additionally, it provides continuous derivatives and relatively low computational cost. In case of tubes, dynamic conditions of tube can not be neglected for accurate simulations. Because of typical high amplification of the tube circuit, the Miller effect [73] must be considered, since it has been affecting the frequency response in audible area. Therefore, a parasitic capacitance  $C_{gp}$  (typically 2.4 pF) should be added into the amplifier circuit model. The other capacitances can be neglected [64].

### 1.4.5 Transformer Core Models

Transformers are inseparable part of many electronic devices, mostly as a part of power supply circuits but it is also possible to find them in the audio signal path. The linear behavior of transformers can be modeled using algorithms described earlier and it has already been discussed in [31, 63]. However, the proper magnetic model must be used to simulate transformers nonlinear behavior expressed using well known relation of magnetic properties  $B = \mu H$ . Since the nonlinear modeling of transformers has become important in the research field of power electronics e.g. [74], several models have been designed [75, 76, 77, 78, 79, 80].

The Frohlich equation used in [74] provides a simple approximation of the transformer core permeability and it is given by

$$B = \frac{H}{c + b|H|} \quad (1.114)$$

where  $c$ ,  $b$  are parameters derived from material properties. The Frohlich model, however, does not support simulation of the hysteresis loop.

When simulating the hysteresis, one can use e.g. Jiles-Atherton (JA) model [81] improved in [82] by removing nonphysical minor hysteresis loops. Magnetization  $M$

of the core is obtained from

$$\frac{dM}{dH} = \delta_M \frac{M_{\text{an}} - M}{k} \text{sign}(dH/dt) + c \frac{dM_{\text{an}}}{dH} \quad (1.115)$$

where  $M_{\text{an}}$  is anhysteretic curve given by Langevin function

$$M_{\text{an}} = M_s \left( \coth \left( \frac{H + \alpha M}{a} \right) - \frac{a}{H + \alpha M} \right). \quad (1.116)$$

Parameters  $M_s$ ,  $\alpha$ ,  $a$ ,  $c$  and  $k$  are derived from material properties and their identification can be found e.g. in [83, 84]. Parameter  $\delta_M = 0$  when the nonphysical minor loop is going to be generated alternatively  $\delta_M = 1$  (refer [82] for details). Finally, the flux density is obtained from

$$B = \mu_0 (M + H). \quad (1.117)$$

Although JA model provides good accuracy, due to its computational cost it might not be convenient for the real-time processing. The efficient model of the nonlinear magnetic core based on the gyrator-capacitor model was introduced in [85]. The saturation effect is simulated by a nonlinear capacitor model and the hysteresis effect using a nonlinear resistor. The nonlinear capacitor is equivalent to a constant capacitor  $C$  in series with a voltage controlled voltage source defined as

$$v(v_C) = |av_C|^n \text{sign}(v_C) \quad (1.118)$$

where  $v_C$  is controlling voltage over the constant capacitor and  $a$ ,  $n$  are saturation parameters. The nonlinear resistor is implemented as a constant resistor  $R$  connected in parallel with the voltage controlled current source defined as

$$i(v_R) = \left| \frac{bv_R}{R} \right|^n \text{sign}(v_R) \quad (1.119)$$

where  $v_R$  is the controlling voltage over the constant resistor and  $b$ ,  $m$  are hysteresis parameters. Because the gyrator is used to transform magnetic properties to electric ones, the current  $i$  corresponds to flux-rate change  $d\Phi/dt$  and the voltage  $v$  corresponds to Magneto-Motive Force (MMF)  $\mathcal{F}$  [86]. The graphic companion circuit as well as the derivation of parameter values can also be found in paper [85]. This transformer core model was successfully used in [46] in the real-time simulation of the audio transformer with one sample delayed nonlinear functions. A slightly different version of functions (1.118),(1.119) was introduced and further, the determination of core parameters from the transformer measurement was derived.



## 2 GOALS OF THESIS

Despite of the recent significant progress in the field of real-time simulation of the analog audio effect, few problems still remain. Foremost, all mentioned algorithms were tested on relatively simple audio circuits. Therefore, the overall goal of this thesis is to investigate possibilities of simulation of the complex nonlinear audio effect circuits in real-time, which will be the guitar amplifier in this case. The block diagram of the guitar amplifier is given in Figure 2.1 [14, 15, 87]. The preamp consists of several triode amplifier stages or limiter units, built by operational amplifiers and diodes, and the passive gain and the tone stack circuit. The power amplifier consists of the phase splitter and the push-pull amplifier built by two pentodes and the output transformer.

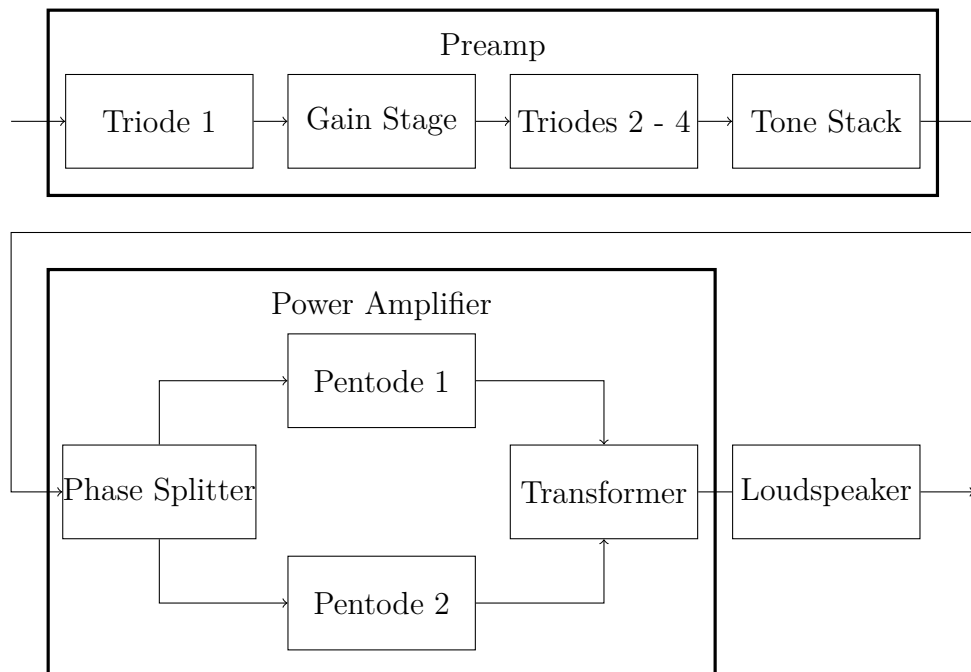


Figure 2.1: Block diagram of guitar amplifier

The DK-method with incidence matrices was chosen as the perspective method for the simulation of more complex circuits, because the second candidate, nonlinear wave digital filters, is not very convenient for the simulation of circuits with more nonlinear functions although it is possible. An extension of DK-method with incidence matrices will be made by introducing the nonlinear model of the audio transformer and also the nonlinear model of the operational amplifier in the first part. This part will also serve as a base for further more complex systems discussed in later parts. Firstly, circuits with the nonlinear transformer model and operational amplifiers will be discussed. On given circuits, several nonlinear models will be investigated

with regards to accuracy and computational complexity and the proper model to be integrated into the DK-method will be chosen.

Because the approximation of system nonlinearities plays the key role in the resulting efficiency of the algorithm, the focus of the next part will be on several approximation algorithms. Firstly, an offline precomputation of the system nonlinearities will be discussed as a way of reduction of the computational complexity. The precomputation of more complex system may lead to large sets of data to be stored in the memory. Therefore, the approximation algorithms will be investigated with regards to computational complexity as well as to the amount of data stored in look-up tables.

The third part of research is focused on the simulation of complex systems. Because the simulation of the complex system can be computationally demanding, techniques of decomposition into simpler parts preserving mutual interaction between blocks will be discussed.

The final part will be based on listening tests. The validity of designed algorithm for simulation of complex systems will be tested. Finally, the presence of aliasing distortion in output signals will be investigated. On the basis of these findings, the minimal required sampling frequency will be stated.

## 3 CIRCUIT ANALYSIS OF AUDIO EFFECTS

The derivation of circuit equations is always the first step in simulation of the circuit. It can be done manually or this process can be automated as has already been mentioned in chapter 1. While the automated process of derivation of circuit equations is very convenient for complex systems, the manual derivation allows better understanding of what happens in the circuit and therefore, it can serve as the initial step for the introduction of new component models into the later automated process of the derivation of circuit equations. It was also shown that manual derivation of circuit equations can produce more efficient computational algorithms, especially for circuits with parametric components [67].

This work continues in prior work of the automated derivation of circuit equations proposed by Yeh [22] and especially Holters and Zölzer [66] whose method of incidence component matrices for derivation of the state-space model of the simulated system will be used. Both works [22, 66] deal with circuits with basic elements. However, several other components, especially the model of the audio transformer as well as the model of Operational Amplifier (OPA), should be considered to be able to simulate audio effects circuits more generally.

Therefore, this chapter will deal with the simulation of circuits containing these types of components. Firstly, the manual derivation of circuit equations will be made. Then, different types of nonlinear models of these components will be investigated with regards to the accuracy of the simulation and the computational cost and finally, the incorporation of these models into the automated process of derivation of the circuit equations will be designed.

### 3.1 Simulation of Circuits with Audio Transformer

Transformers can be found in many audio effect or audio amplifier circuits. However, they are often a part of power supply circuits and thus, they can be mostly neglected in simulations of these electronic devices, or their simulation can be done separately. These transformers can be simulated using the ideal transformer model because power transformers are usually designed to handle sufficient power and therefore they work in the linear area of the magnetization curve.

However, transformers are often connected directly in the audio path in many types of audio circuits, e.g. input and output stages of mixing consoles, professional studio compressors, etc., where transformers are used as asymmetrical-to-symmetrical or symmetrical-to-asymmetrical converters. Similar types of the transformers can also be found in tube power amplifiers where their main function is to match the high output impedance of used tubes to the low input impedance of the loudspeaker.

Although designers of the audio transformers try to minimize the distortion effect of transformers in order to get a transparent signal path, the presence of the transformer can be often heard and sometimes it is even required by musicians and audio engineers. The main reason for this is the characteristic nonlinear distortion caused by the soft saturation of the transformer core when a large magnetization occurs. However, the saturation effect is not the only one which manifests in the output signal, the hysteresis effect and frequency properties also have to be included in the simulation. The transformer model can be split into three parts – the electric part modeling parasitic capacitances of windings as well as leakage inductances, the electromagnetic part responsible for interaction of electric and magnetic field and the magnetic part which models the core saturation, the hysteresis and eddy currents.

### 3.1.1 Transformer Model

There are three approaches to the transformer modeling – the ideal model, the linear model and the nonlinear model of the transformer. The ideal transformer with number of winding turns  $N_1, N_2$  is considered to be an impedance divider that transforms input voltages  $v_p$  and currents  $i_p$  to output  $v_s$  and  $i_s$  according to

$$\frac{v_s}{v_p} = \frac{N_s}{N_p} = \frac{i_p}{i_s} \quad (3.1)$$

and although the ideal transformer is far away from the real one, it can be an useful model from the real-time simulation point of view. However, for more accurate simulation, losses caused by the hysteresis and the core saturation have to be considered as well as parasitic capacitances of windings, leakage inductances and windings resistances.

The model of the transformer with two windings derived from [46] is shown in Figure 3.1. However, the resistances  $R_1, R_2$ , the parasitic capacitances  $C_1, C_2$  and the leakage inductances  $L_1, L_2$  are all modeled in the electric part of the model which is in contrast with [46] where the leakage inductance is modeled in the magnetic domain. Because the electric part of transformer can be easily incorporated directly into the whole circuit, if necessary, these electric components are no further considered as a part of the transformer model but rather as a part of the whole circuit.

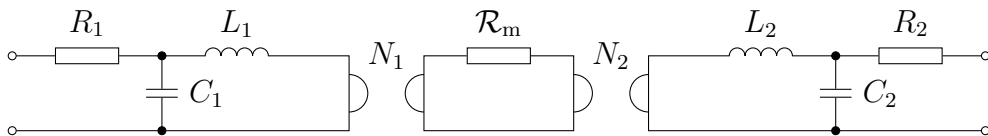


Figure 3.1: Model of a transformer with two windings.

The electromagnetic part of the transformer model is made by two gyrators with resistances corresponding to numbers of turns of the windings  $N_1, N_2$ . However, both gyrators serve only as a graphical representation of the electromagnetic interaction in the model 3.1 and they will not be used during derivation of the model equations. The electromagnetic interaction will be modeled using Amper's law

$$\oint H dl = \sum_i I_i \quad (3.2)$$

which is for homogeneous magnetizing force  $H$ , number of turns  $N_i$  of  $i$ th winding and the length of the magnetic path  $l_{\text{mag}}$  in form

$$H l_{\text{mag}} = \sum_i N_i I_i = \mathcal{F} \quad (3.3)$$

where  $\mathcal{F}$  is MMF and further Faraday's induction law for homogeneous magnetic flux  $\Phi$

$$v_i = N_i \frac{d\Phi}{dt} = N_i S \frac{dB}{dt} \quad (3.4)$$

where  $S$  is the transformer core cross-section area and  $B$  is the flux density.

The magnetic part of the model can be either linear or nonlinear. In case of the linear magnetic core, the relation between magneto-motive force  $\mathcal{F}$  and flux  $\Phi$  is given by Hopkinson's law

$$\mathcal{F} = \mathcal{R} \Phi \quad (3.5)$$

where  $\mathcal{R}$  is the magnetic reluctance (denoted in Figure 3.1 as  $\mathcal{R}_m$ ) and given by

$$\mathcal{R} = \frac{l_{\text{mag}}}{\mu S}. \quad (3.6)$$

In case of the nonlinear core model, the reluctance  $\mathcal{R}$  is nonlinear because of nonlinear permeability  $\mu$ . To simulate the nonlinear core behavior, one can use the models given in chapter 1.4.5. We can reformulate all model equations in terms of  $\mathcal{F}$  and  $\Phi$  variables. The Frohlich model can be expressed as

$$\mathcal{F}(\Phi) = \frac{\frac{c l_{\text{mag}}}{S} \Phi}{1 - \frac{b}{S} |\Phi|} = \frac{\bar{c} \Phi}{1 - \bar{b} |\Phi|} \quad (3.7)$$

or

$$\Phi(\mathcal{F}) = \frac{\mathcal{F}}{\bar{c} + \bar{b} |\mathcal{F}|} \quad (3.8)$$

where  $\bar{c} = \frac{c l_{\text{mag}}}{S}$  and  $\bar{b} = \frac{b}{S}$  and parameters  $c, b$  are given by [74]

$$c = \frac{1}{\mu_i \mu_0} = \frac{1}{\mu}, \quad (3.9)$$

$$b = \frac{1 - \frac{1}{\sqrt{\mu_i}}}{B_{\text{SAT}}}. \quad (3.10)$$

The saturation flux density  $B_{\text{SAT}}$  and the permeability  $\mu_i$  enable modeling of different types of the magnetic core. This allows designing transformer models from the material datasheet and arbitrary transformer dimensions. However, data of measured transformers are often provided in terms of  $\Phi$  and  $\mathcal{F}$  and actual dimensions of transformer are not specified. Hence, it would be useful to derive parameters of Frohlich model from measured data. Parameters  $\bar{c}$ ,  $\bar{b}$  used in (3.7),(3.8) can be expressed as

$$\bar{c} = \frac{cl_{\text{mag}}}{S} = \frac{l_{\text{mag}}}{\mu S} = \mathcal{R} \quad (3.11)$$

where  $\bar{c}$  is equal to the core reluctance in the linear part of the transfer function and parameter  $b$

$$\bar{b} = \frac{b}{S} = \frac{1 - \frac{1}{\sqrt{\frac{1}{\bar{c}\mu_0}}}}{\Phi_{\text{SAT}}}. \quad (3.12)$$

It is possible to derive both parameters from the estimated magnetic reluctance and saturation flux but better results can be obtained by using LMS parameter fitting algorithm to derive model parameters.

The magnetic reluctance of the gyrator-capacitor transformer model [46] is modeled with the nonlinear capacitance and the resistance which are implemented as the constant capacitor with capacity  $C$  given by the magnetic permeance  $C = 1/\mathcal{R}$  with the voltage source controlled by the voltage over the constant capacitor given by (1.118) and the parallel resistor and the current source controlled by the voltage over the resistor given by (1.119). The details are shown in Figure 3.2.

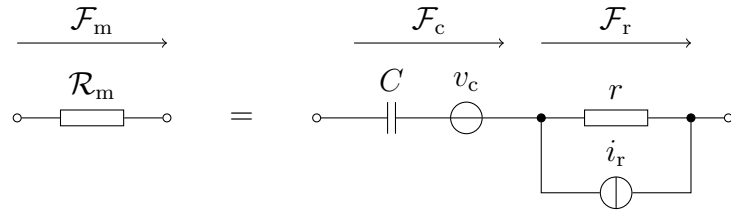


Figure 3.2: Magnetic part of gyrator-capacitor transformer model.

Because magnetic circuits obey similar rules as electric circuits [86], it is possible to write the relation  $\sum_i \mathcal{F}_i = 0$  for the magnetic circuit loop and hence total magnetomotive force over magnetic reluctance  $\mathcal{R}_m$  is

$$\mathcal{F}_m = \mathcal{F}_c + \mathcal{F}_r. \quad (3.13)$$

Using equations (1.118), (1.119) and the constant capacitor  $C$  and resistor  $r$  it is possible to derive

$$\mathcal{F}_c = \frac{\Phi}{C} + a \left| \frac{\Phi}{C} \right|^n \text{sign}(\Phi) \quad (3.14)$$

$$\frac{d\Phi}{dt} = \frac{\mathcal{F}_r}{r} + b \left| \frac{\mathcal{F}_r}{r} \right|^m \text{sign}(\mathcal{F}_r) \quad (3.15)$$

where the term  $\frac{d\Phi}{dt}$  corresponds to the current in gyrator-capacitor approach of modeling magnetic circuits. To express the total magneto-motive force  $\mathcal{F}_m$  both equations (3.14),(3.15) are combined. Unfortunately, because of the form of equation (3.15), the variable  $\mathcal{F}_r$  cannot be isolated analytically and it forms only the expression

$$\mathcal{F}_r = \frac{\frac{d\Phi}{dt} - b \left| \frac{\mathcal{F}_r}{r} \right|^m \text{sign}(\mathcal{F}_r)}{r}. \quad (3.16)$$

The total magneto-motive force  $\mathcal{F}_m$  is then

$$\mathcal{F}_m = \frac{\frac{d\Phi}{dt} - b \left| \frac{\mathcal{F}_r}{r} \right|^m \text{sign}(\mathcal{F}_r)}{r} + \frac{\Phi}{C} + a \left| \frac{\Phi}{C} \right|^n \text{sign}(\Phi) \quad (3.17)$$

for

$$\mathcal{F}_r = \sum_i N_i I_i - \mathcal{F}_c \quad (3.18)$$

which can be computed only numerically for  $\mathcal{F}_m = \sum_i N_i I_i$ . However, it is more practical to approximate  $\mathcal{F}_r$  as a function  $\mathcal{F}_r(\frac{d\Phi}{dt})$  which gives

$$\mathcal{F}_m = \mathcal{F}_c(\Phi) + \mathcal{F}_r \left( \frac{d\Phi}{dt} \right) = \mathcal{F} \left( \Phi, \frac{d\Phi}{dt} \right) \quad (3.19)$$

as a function of two variables  $\Phi$  and  $\frac{d\Phi}{dt}$ . The identification of parameters from datasheet data is given in [85] and from measured data in [46].

The third core model considered in this thesis is the JA core model given by equations (1.115), (1.116) and (1.117). The JA model is defined for variables  $H$ ,  $B$  but after derivation of the model it can be transformed into  $\mathcal{F}$ ,  $\Phi$  space with substitution  $\mathcal{F} = H l_{\text{mag}}$  and  $\Phi = BS$ . Firstly, the derivative  $\frac{dM}{dH}$  in (1.115) is replaced by time derivatives  $\frac{dM}{dt}$  and  $\frac{dH}{dt}$

$$\frac{dM}{dH} = \delta \frac{M_{\text{an}} - M}{k} + c \frac{dM_{\text{an}}}{dH} \quad (3.20)$$

$$\frac{dM}{dt} \frac{dt}{dH} = \delta \frac{M_{\text{an}} - M}{k} + c \frac{dM_{\text{an}}}{dt} \frac{dt}{dH} \quad (3.21)$$

$$\frac{dM}{dt} = \delta \frac{M_{\text{an}} - M}{k} \frac{dH}{dt} + c \frac{dM_{\text{an}}}{dt} \quad (3.22)$$

where  $\delta = \delta_M \text{sign} \left( \frac{dH}{dt} \right)$ . Subsequently, the magnetization  $M$  can be substituted by

$$M = \left( \frac{B}{\mu_0} - H \right) \quad (3.23)$$

derived from (1.117) to get

$$\frac{d\left(\frac{B}{\mu_0} - H\right)}{dt} = \delta \frac{M_{\text{an}} - \frac{B}{\mu_0} - H}{k} \frac{dH}{dt} + c \frac{dM_{\text{an}}}{dt} \quad (3.24)$$

from (3.22) and

$$M_{\text{an}} = M_s \left( \coth \left( \frac{H + \alpha \left( \frac{B}{\mu_0} - H \right)}{a} \right) - \frac{a}{H + \alpha \left( \frac{B}{\mu_0} - H \right)} \right) \quad (3.25)$$

from (1.116) where  $H = \frac{\mathcal{F}}{l_{\text{mag}}}$  and  $B = \frac{\Phi}{S}$ . Neither  $H$  resp.  $\mathcal{F}$  nor  $B$  resp.  $\Phi$  can be isolated from JA model (3.24) and therefore it can be expressed only in implicit form

$$0 = f_{\text{JA}} \left( H, B, \frac{dH}{dt}, \frac{dB}{dt} \right). \quad (3.26)$$

### 3.1.2 Basic Input Stage with Transformer

Transformer models cannot be compared only from their equations but they must be incorporated into a circuit. For this purpose, a simple circuit, which can be found as an input stage of most of professional audio equipment, was investigated in [4, 5]. The input stage circuit contains a transformer used as a converter from a symmetrical to an asymmetrical signal. This type of audio transformer consists of two primary windings and one secondary winding. The circuit schematic of the transformer connected with two voltage sources, which represent the symmetrical signal, is shown in Figure 3.3. The resistors  $R_{p1}$  and  $R_{p2}$  represent the primary coil resistance, the resistor  $R_s$  is the series combination of the secondary coil resistance and the load resistance. Symbols  $N_{p1}$ ,  $N_{p2}$  and  $N_s$  are numbers of turns of the windings. The leakage inductance as well as the capacitance of windings are neglected in this model but they can be easily added according to Figure 3.1. However, audio transformers are often designed in that way that the influence of the leakage inductance and the capacitance does not manifest in the audible region [88].

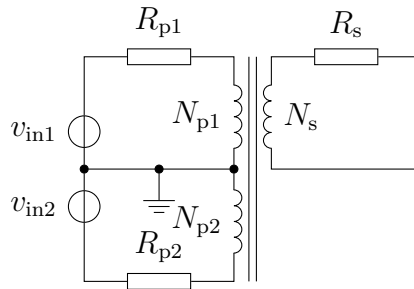


Figure 3.3: A transformer connected with a symmetrical voltage signal source.



The circuit in Figure 3.3 contains five unknown variables –  $v_{p1}$ ,  $v_{p2}$ ,  $v_s$ ,  $\Phi$  and  $\mathcal{F}$ . Therefore, the following five equations are necessary for the solution of the system

$$\mathcal{F} = N_{p1} \frac{v_{in1} - v_{p1}}{R_{p1}} - N_{p2} \frac{v_{in2} + v_{p2}}{R_{p2}} - N_s \frac{v_s}{R_s}, \quad (3.27)$$

$$\frac{d\Phi}{dt} = \frac{v_{p1}}{N_{p1}}, \quad (3.28)$$

$$\frac{d\Phi}{dt} = \frac{v_{p2}}{N_{p1}}, \quad (3.29)$$

$$\frac{d\Phi}{dt} = \frac{v_s}{N_s}, \quad (3.30)$$

$$\mathcal{F} = \mathcal{R}\Phi. \quad (3.31)$$

The first equation is obtained from (3.3) where currents  $I_{p1}$  and  $I_{p2}$  are computed using Ohm's law from input voltages  $v_{in1}$  and  $v_{in2}$ . Equations (3.28), (3.29) and (3.30) are obtained using Faraday's law for different voltages and equation (3.31) describes the transformer core behavior. If the model operates with small signals, then reluctance  $\mathcal{R}$  is considered as a constant and the whole system is linear. However, it varies with magneto-motive force  $\mathcal{F}$  and flux  $\Phi$  for large signals and models from previous chapter can be used instead of (3.31). The voltages  $v_{p1}$ ,  $v_{p2}$ ,  $v_s$  are linearly dependent and as a result it is possible to simplify the equations to

$$\mathcal{F} = N_{p1} \frac{v_{in1} - N_{p1}v}{R_{p1}} - N_{p2} \frac{v_{in2} + N_{p2}v}{R_{p2}} - N_s \frac{N_s v}{R_s}, \quad (3.32)$$

$$\frac{d\Phi}{dt} = v, \quad (3.33)$$

$$\mathcal{F} = \mathcal{R}\Phi. \quad (3.34)$$

To implement the transformer model in the digital domain, the differential equation (3.33) has to be discretized. Using BE discretization, we can get

$$\frac{\Phi[n] - \Phi[n-1]}{T} = v \quad (3.35)$$

where  $n$  denotes the time index and  $T$  is the sample period.

The final equations for the Frohlich model are

$$\begin{aligned} \mathcal{F} &= N_{p1} \frac{v_{in1} - N_{p1}v}{R_{p1}} - N_{p2} \frac{v_{in2} + N_{p2}v}{R_{p2}} - N_s \frac{N_s v}{R_s} \\ v &= \frac{\Phi[n] - \Phi[n-1]}{T} \\ \mathcal{F} &= \frac{\bar{c}\Phi}{1 - \bar{b}|\Phi|}. \end{aligned} \quad (3.36)$$

The equations for the gyrator-capacitor model are

$$\begin{aligned}
\mathcal{F} &= N_{p1} \frac{v_{in1} - N_{p1}v}{R_{p1}} - N_{p2} \frac{v_{in2} + N_{p2}v}{R_{p2}} - N_s \frac{N_s v}{R_s} \\
v &= \frac{\Phi[n] - \Phi[n-1]}{T} \\
v &= \frac{\mathcal{F}_r}{r} + b \left| \frac{\mathcal{F}_r}{r} \right|^m \text{sign}(\mathcal{F}_r) \\
\mathcal{F} &= \mathcal{F}_r + \frac{\Phi}{C} + a \left| \frac{\Phi}{C} \right|^n \text{sign}(\Phi).
\end{aligned} \tag{3.37}$$

And finally equations for the JA model are

$$\begin{aligned}
\mathcal{F}[n] &= N_{p1} \frac{v_{in1} - N_{p1}v}{R_{p1}} - N_{p2} \frac{v_{in2} + N_{p2}v}{R_{p2}} - N_s \frac{N_s v}{R_s} \\
v &= \frac{\Phi[n] - \Phi[n-1]}{T} \\
0 &= f_{JA} \left( \frac{\mathcal{F}[n]}{l_{mag}}, \frac{\Phi}{S}, \frac{\mathcal{F}[n] - \mathcal{F}[n-1]}{T l_{mag}}, \frac{v}{S} \right).
\end{aligned} \tag{3.38}$$

All the circuit equations with different transformer models require numerical algorithm to solve. Equations (3.36) contain one unknown variable  $\Phi$  and equations (3.37) and (3.38) contain unknown  $\Phi$  and  $\mathcal{F}_r$  resp.  $\mathcal{F}$ . It is also possible to express the equations in terms of only unknown  $\Phi$  but problems with numerical stability occurred in this case. Values of circuit components as well as parameters for all transformer models are listed in Tables 3.1, 3.2 and 3.3. the Newton-Raphson algorithm was chosen for numerical solving. The comparison of the computational efficiency can be found in Table 3.4 where the maximum number and the average number of iterations is given for each transformer model.

Table 3.1: Values of circuit components of input stage circuit.

$R_{p1}$	$R_{p2}$	$R_s$	$N_{p1}$	$N_{p2}$	$N_s$
100 $\Omega$	100 $\Omega$	1 k $\Omega$	100	100	100

Table 3.2: Parameter of Frohlich and GC-model of transformer core.

Frohlich			GC-model				
$\bar{b}$	$\bar{c}$	$C$	$a$	$n$	$r$	$b$	$m$
255	358	$2 \cdot 10^{-3}$	$1 \cdot 10^{-5}$	25	2.3	8.4	4

Table 3.3: Parameter of JA-model of transformer core.

$M_s$	$a$	$\alpha$	$c$	$k$	$l_{\text{mag}}$	$S$
$9.9 \cdot 10^5$	35	$9 \cdot 10^{-5}$	$4.6 \cdot 10^{-2}$	26	$2.7 \cdot 10^{-3}$	$3 \cdot 10^{-3}$

Table 3.4: Number of iterations of Newton-Raphson method.

Model	Frohlich	GC-model	JA-model
Max	5.00	4.00	53.00
Average	2.93	2.96	12.58

Figure 3.4 shows the output signal comparison for all transformer models. The input sine wave signal with a frequency of 30 Hz and an amplitude of 150 V was used. This signal was used to set the transformer core into full saturation. The hysteresis loop for all transformer models is shown in Figure 3.5.

Results have confirmed prior assumptions. The JA-model provides good modeling of the hysteresis loop but on the other hand, the computational cost is very high and problems with numerical stability can also occur foremost in the area where  $\Phi$  and  $\mathcal{F}$  are close to zero. The Frohlich model and GC model have almost the same computational cost (computational cost of GC-model is higher due to the more complex model equation) but the advantage is modeling of the hysteresis loop.

System equations can also be expressed in terms of the DK-method, which is useful for the offline precomputation of nonlinear equations when there are more system states than the number of nonlinear equations. Substitution of (3.35) into (3.32) leads to

$$\begin{aligned} \mathcal{F}[n] = & \left( \frac{N_{p1}^2}{R_{p1}} - \frac{N_{p2}^2}{R_{p2}} - \frac{N_s^2}{R_s} \right) \frac{\Phi[n] - \Phi[n-1]}{T} + \\ & + \frac{N_{p1} v_{in1}[n]}{R_{p1}} - \frac{N_{p2} v_{in2}[n]}{R_{p2}} \end{aligned} \quad (3.39)$$

where magneto-motive force  $\mathcal{F}$  is the input  $\mathbf{v}$  of the nonlinear function, the vector of state variables can be given according to

$$\mathbf{x}[n-1] = [\Phi[n-1]] \quad (3.40)$$

and the input vector by

$$\mathbf{u}[n] = \begin{bmatrix} v_{in1}[n] \\ v_{in2}[n] \end{bmatrix}. \quad (3.41)$$

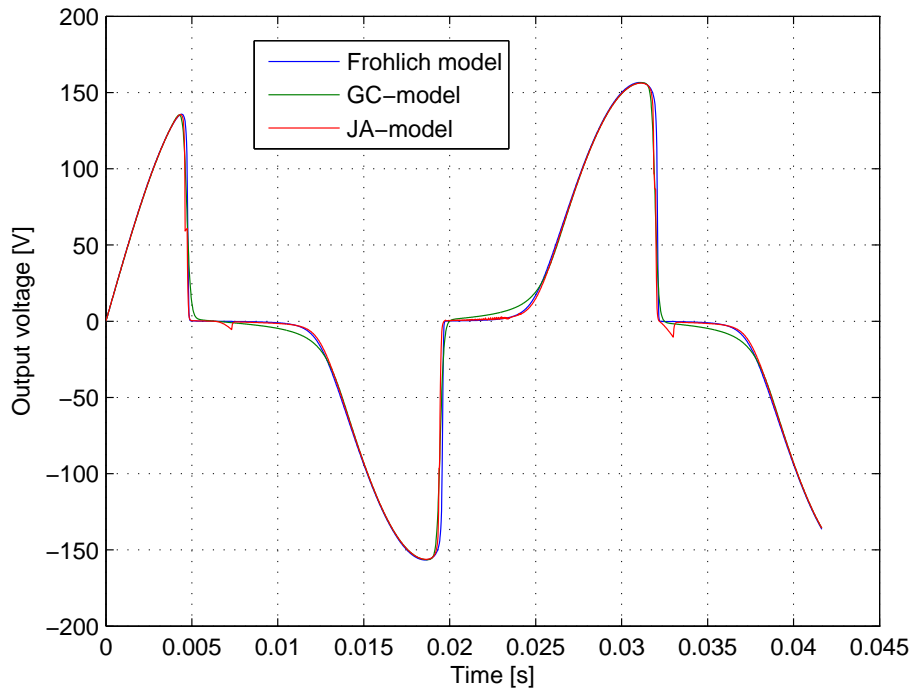


Figure 3.4: Output signal of the input stage circuit.

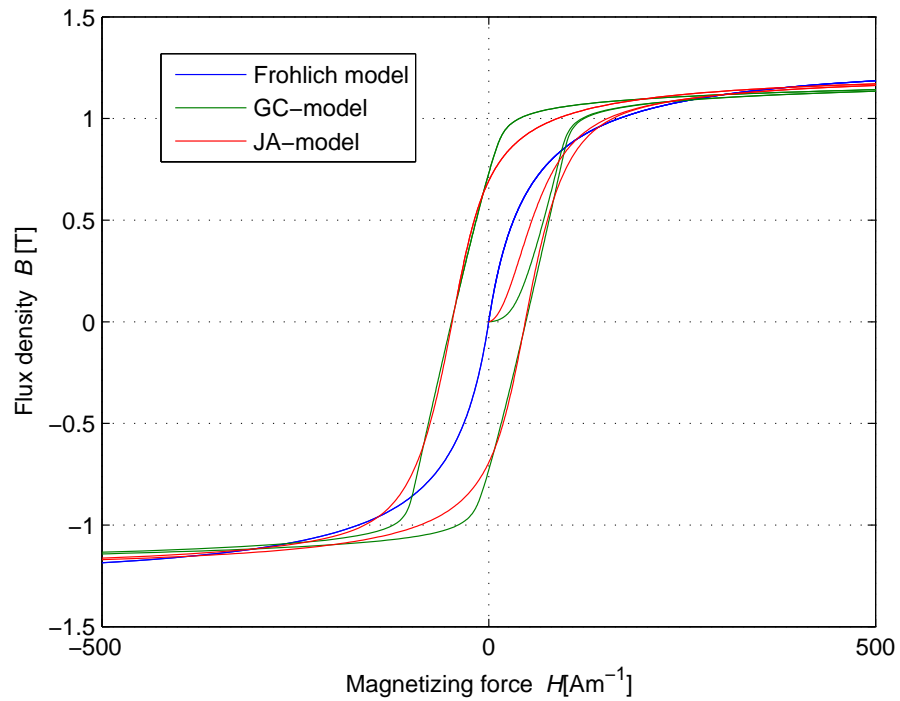


Figure 3.5: Hysteresis loop of transformer models used in input stage circuit.

The coefficient matrices  $\mathbf{G}$ ,  $\mathbf{H}$  and  $\mathbf{K}$  are derived by comparing equations (3.39) and (1.61) as

$$\mathbf{G} = \left[ \left( -\frac{N_{p1}^2}{R_{p1}} + \frac{N_{p2}^2}{R_{p2}} + \frac{N_s^2}{R_s} \right) \frac{1}{T} \right], \quad (3.42)$$

$$\mathbf{H} = \left[ \begin{array}{cc} \frac{N_{p1}}{R_{p1}} & -\frac{N_{p2}}{R_{p2}} \end{array} \right], \quad (3.43)$$

$$\mathbf{K} = \left[ \left( \frac{N_{p1}^2}{R_{p1}} - \frac{N_{p2}^2}{R_{p2}} - \frac{N_s^2}{R_s} \right) \frac{1}{T} \right]. \quad (3.44)$$

The coefficient matrices  $\mathbf{A}$ ,  $\mathbf{B}$  and  $\mathbf{C}$  are derived by comparing (3.35) and (1.59) as

$$\mathbf{A} = [0], \quad (3.45)$$

$$\mathbf{B} = \left[ \begin{array}{cc} 0 & 0 \end{array} \right], \quad (3.46)$$

$$\mathbf{C} = [1], \quad (3.47)$$

and coefficient matrices  $\mathbf{D}$ ,  $\mathbf{E}$  and  $\mathbf{F}$  are obtained using (3.30) and (1.60) as

$$\mathbf{D} = \left[ -\frac{N_s}{T} \right], \quad (3.48)$$

$$\mathbf{E} = \left[ \begin{array}{cc} 0 & 0 \end{array} \right], \quad (3.49)$$

and

$$\mathbf{F} = \left[ \frac{N_s}{T} \right]. \quad (3.50)$$

The final nonlinear equation is then given by

$$0 = \mathbf{G}\mathbf{x} + \mathbf{H}\mathbf{u} + \mathbf{K} \cdot \Phi(\mathcal{F}[n]) - \mathcal{F}[n] = p + K \cdot \Phi(\mathcal{F}[n]) - \mathcal{F}[n]. \quad (3.51)$$

The nonlinear function is given in  $\Phi(\mathcal{F}[n])$ . To express the nonlinear equation in terms of  $\mathcal{F}(\Phi[n])$ , the variable  $\Phi$  is isolated from (3.39) resulting in

$$\begin{aligned} \Phi[n] = & \Phi[n-1] + \frac{\mathcal{F}[n]}{\left( -\frac{N_{p1}^2}{R_{p1}T} - \frac{N_{p2}^2}{R_{p2}T} + \frac{N_s^2}{R_sT} \right)} \\ & + \frac{1}{\left( -\frac{N_{p1}^2}{R_{p1}T} - \frac{N_{p2}^2}{R_{p2}T} + \frac{N_s^2}{R_sT} \right)} \left( -\frac{N_{p1}}{R_{p1}} v_{in1}[n] + \frac{N_{p2}}{R_{p2}} v_{in2}[n] \right) \end{aligned} \quad (3.52)$$

and thus, matrices  $\mathbf{G}$ ,  $\mathbf{H}$  and  $\mathbf{K}$  are following

$$\mathbf{G} = [1], \quad (3.53)$$

$$\mathbf{H} = \begin{bmatrix} \frac{-N_{p1}}{\left(-\frac{N_{p1}^2}{R_{p1}T} - \frac{N_{p2}^2}{R_{p2}T} + \frac{N_s^2}{R_sT}\right)R_{p1}} & \frac{N_{p2}}{\left(-\frac{N_{p1}^2}{R_{p1}T} - \frac{N_{p2}^2}{R_{p2}T} + \frac{N_s^2}{R_sT}\right)R_{p2}} \end{bmatrix}, \quad (3.54)$$

$$\mathbf{K} = \begin{bmatrix} \frac{1}{\left(-\frac{N_{p1}^2}{R_{p1}T} - \frac{N_{p2}^2}{R_{p2}T} + \frac{N_s^2}{R_sT}\right)} \end{bmatrix} \quad (3.55)$$

and the nonlinear equation to be solved is

$$0 = \mathbf{G}\mathbf{x} + \mathbf{H}\mathbf{u} + \mathbf{K} \cdot \mathcal{F}(\Phi[n]) - \Phi[n] = p + K \cdot \mathcal{F}(\Phi[n]) - \Phi[n]. \quad (3.56)$$

The other matrices remain the same. The solved nonlinear function (3.56) for Frohlich and GC-model is shown in Figures 3.6 and 3.7. The solution for the Frohlich model depends only on one input variable  $p$  and can be easily approximated (as will be discussed later) to reduce the computational complexity of the simulation algorithm. The solution for the GC-model further depends on the state variable  $\Phi[n-1]$  and therefore it requires two dimensional approximation. And finally, three input variables  $p$ ,  $\Phi[n-1]$  and  $\mathcal{F}[n-1]$  are needed for the JA-model.

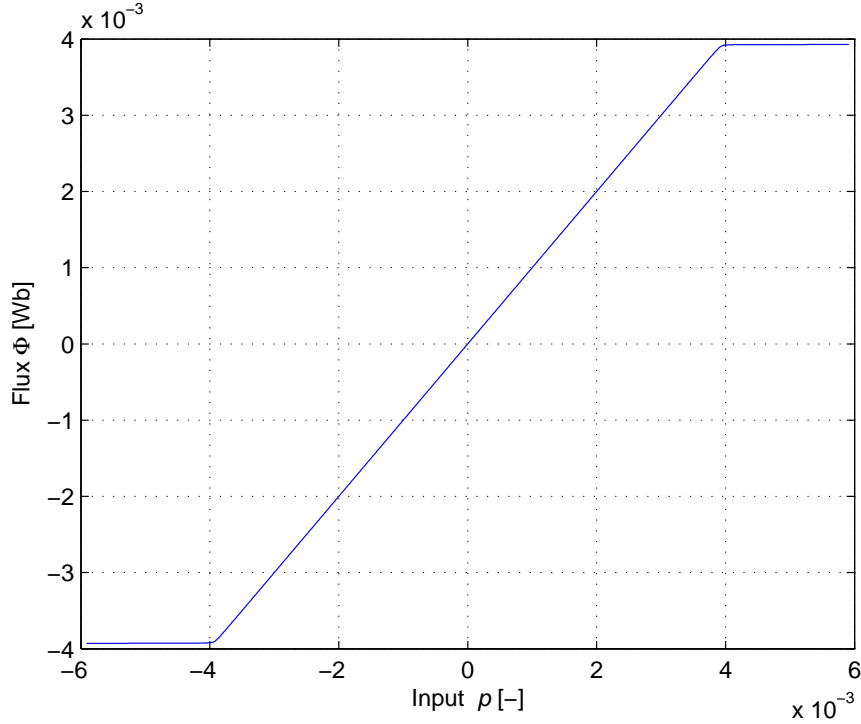


Figure 3.6: Solution of input stage nonlinear function with Frohlich model.

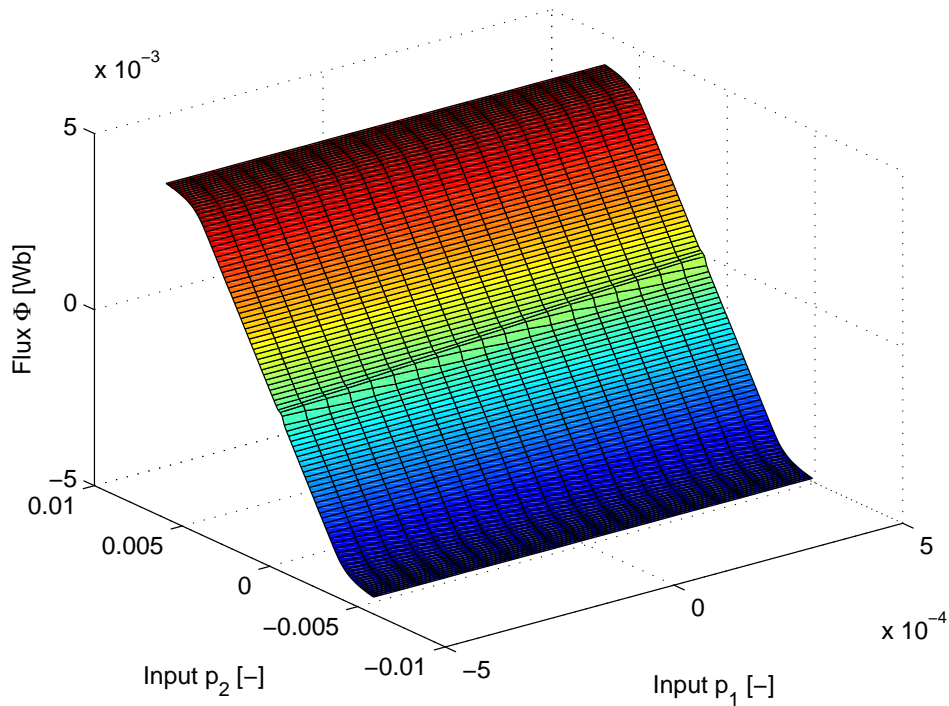


Figure 3.7: Solution of input stage nonlinear function with the GC model.

### 3.1.3 Push-Pull Tube Amplifier

The second commonly used audio circuit with the audio transformer is a push-pull tube amplifier. The simulation of the push-pull tube amplifier with different transformer models was published in [6]. The complete push-pull amplifier consists of several parts – the phase splitter, two or four power tubes which processes opposite half-waves of the signal and the output transformer which sums contributions from opposite power tubes and also a model of a loudspeaker must be included because the loudspeaker is a frequency dependent load. However, the simulation of the complete push-pull power amplifier will be described later and only the simulation of the power tubes part with the output transformer will be considered in this chapter. The simplified circuit schematic of the power tubes part is shown in Figure 3.8 and values for circuit elements are listed in Table 3.5. Values were chosen according to Fender Deluxe Reverb power amplifier [89]. The circuit consist of two 6V6GT pentodes fed by phase inverted input signals and a three windings transformer. The loudspeaker is modeled only by the resistor because the output transformer is the main objective here. The output transformer type is Fender NSC041318. This transformer was measured in [46] and transformer parameters (see Table 3.6) can be found there as well. Unfortunately, the exact number of turns per winding is unknown and only the

transformer ratio is given. The number of turns per winding has been here chosen as  $N_1 = 1000$ ,  $N_2 = 1000$  and  $N_3 = 64$ , which are more typical in real amplifiers than values given in [46].

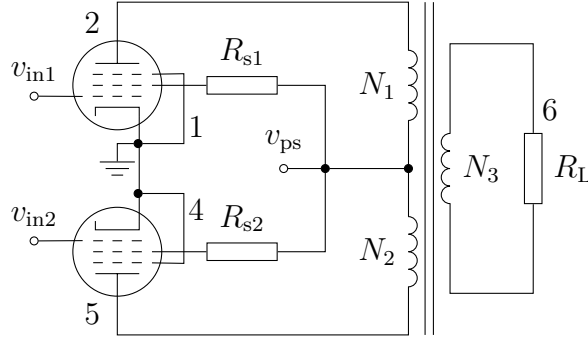


Figure 3.8: Circuit schematic for push-pull power tubes part of the tube power amplifier.

Table 3.5: Values for circuit elements for push-pull power tubes part of the tube power amplifier.

$R_{s1}$	$R_{s2}$	$R_L$	$N_1$	$N_2$	$N_3$	$V_{ps}$
$470 \Omega$	$470 \Omega$	$8 \Omega$	1000	1000	64	394 V

Table 3.6: Parameter of Frohlich and GC-model of the transformer core for the transformer Fender NSC041318.

$C$	$a$	$n$	$r$	$b$	$m$
$24 \cdot 10^{-3}$	900	7	$7.7 \cdot 10^{-2}$	4.5	4

The circuit from Figure 3.8 can be described by equations

$$G_{s1}(v_{ps} - v_1) - I_s(v_{in1}, v_1) = 0, \quad (3.57)$$

$$G_{s2}(v_{ps} - v_4) - I_s(v_{in2}, v_4) = 0, \quad (3.58)$$

$$(v_2 - v_{ps}) - N_1 \frac{d\Phi}{dt} = 0, \quad (3.59)$$

$$(v_{ps} - v_5) - N_2 \frac{d\Phi}{dt} = 0, \quad (3.60)$$

$$-\mathcal{F} + N_1 I_p(v_{in1}, v_1, v_2) + N_2 I_p(v_{in2}, v_4, v_5) + N_3^2 \frac{d\Phi}{dt} G_L = 0, \quad (3.61)$$

$$\mathcal{F} = f(\Phi) \quad (3.62)$$



where different transformer core models can be substituted into (3.62) and symbols  $G$  are conductances of the resistors from Table 3.5. After discretization, the system can be solved numerically for unknown  $\Phi$ ,  $v_1$ ,  $v_2$ ,  $v_4$  and  $v_5$ .

Several simulations for different input signals were made to find the effect of the output transformer to the circuit. All input voltages were biased to  $-37$  V because the grid circuit is missing in the circuit from Figure 3.8. Figure 3.9 shows the comparison of output signals of the push-pull amplifier simulation for different transformer models – linear model, GC-model without modeling the hysteresis effect and full GC-model. The input signal was a sine wave with a frequency of 30 Hz and an amplitude of 50 V. One can observe the waveform distortion caused by the magnetic core saturation. The difference between the simulation with the GC-model without modeling the hysteresis and the full GC-model is very subtle and therefore, the GC-model consisting only of the saturation equation (3.14) can be use without loosing the accuracy of the simulation.

Further, the distortion analysis of the push-pull amplifier with the linear and nonlinear output transformer model is shown in Figures 3.10 and 3.11. It can be seen that the model with the nonlinear output transformer adds some distortion in the area of low frequencies (bellow 120 Hz) and therefore it implies that the nonlinear model of the transformer should be used because the distortion could be noticeable. It will be later investigated by listening tests.

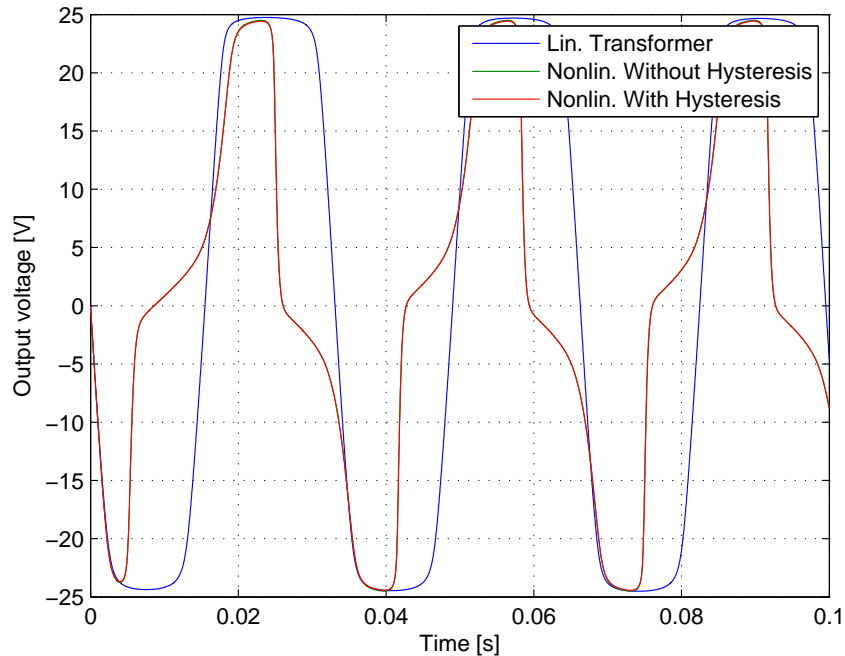


Figure 3.9: Ouput signals of the push-pull amplifier for different transformer models.

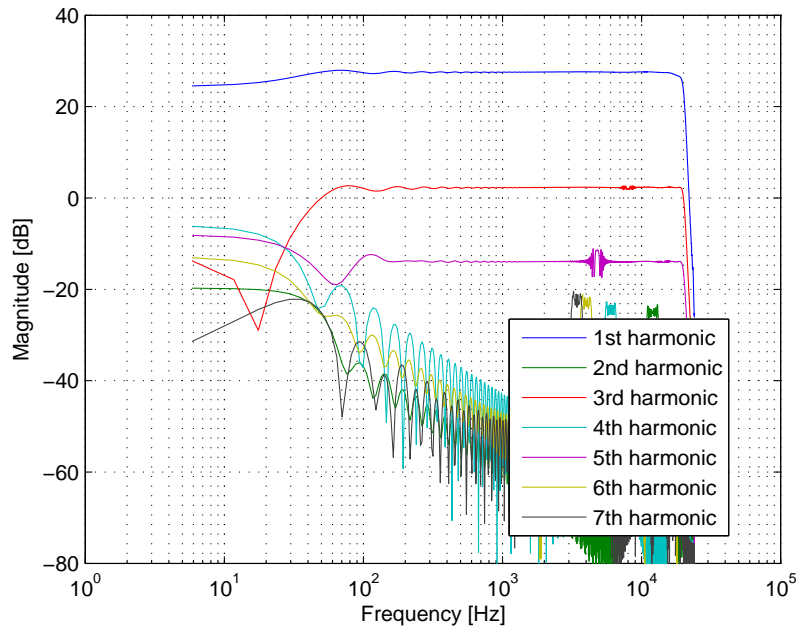


Figure 3.10: Distortion analysis of the push-pull amplifier with the linear model of the output transformer.

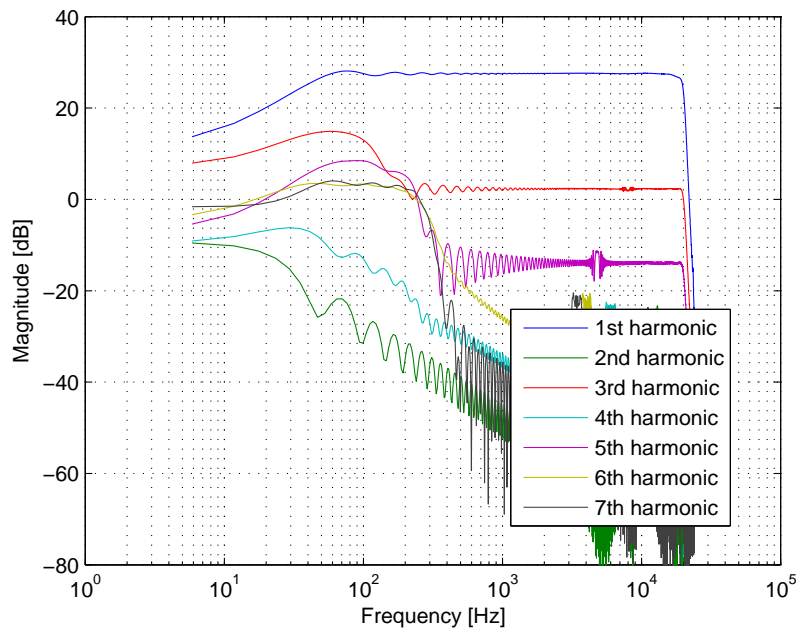


Figure 3.11: Distortion analysis of the push-pull amplifier with the nonlinear model of the output transformer.

### 3.1.4 Automated Incorporation of Transformer Model into DK-method

The derivation of DK-method matrices for the push-pull power amplifier can be done similarly as the derivation of matrices for the input stage circuit in chapter 3.1.2. However, it is more complicated because of more nonlinear equations in the circuit. Therefore, it would be beneficial to derive DK-method matrices automatically. Original equations (3.57)–(3.62) are not convenient for incorporation of the transformer model into automated DK-method and therefore they were reformulated into

$$0 = G_{s1}(v_{ps} - v_1) - I_s(v_{in1}, v_1), \quad (3.63)$$

$$0 = G_{s2}(v_{ps} - v_4) - I_s(v_{in2}, v_4), \quad (3.64)$$

$$0 = (v_2 - v_{ps}) - N_1 \frac{d\Phi}{dt}, \quad (3.65)$$

$$0 = (v_{ps} - v_5) - N_2 \frac{d\Phi}{dt}, \quad (3.66)$$

$$0 = N_3 \frac{d\Phi}{dt} G_L - I_3, \quad (3.67)$$

$$0 = -\mathcal{F} + N_1 I_1 + N_2 I_2 + N_3 I_3, \quad (3.68)$$

$$I_1 = I_p(v_{in1}, v_1, v_2), \quad (3.69)$$

$$I_2 = I_p(v_{in2}, v_4, v_5), \quad (3.70)$$

$$\mathcal{F} = \Phi \mathcal{R}(\Phi). \quad (3.71)$$

The DK-method requires prior component-wise discretization of energy storing circuit components. The Faraday's induction law can be discretized by either TR or BE discretization formula. Using TR, it is possible to derive companion circuit consisting of conductance  $G$  and current source  $x$  according to

$$v = \frac{Nd\Phi}{dt} \quad (3.72)$$

$$\frac{1}{2}(v[n] + v[n-1]) = \frac{N}{T}(\Phi[n] - \Phi[n-1]) \quad (3.73)$$

$$v[n] = \frac{2N}{T}(\Phi[n] - \Phi[n-1]) - v[n-1] \quad (3.74)$$

$$v[n] = \frac{2N}{T}\Phi[n] - x[n-1] \quad (3.75)$$

$$v[n] = G\Phi[n] - x[n-1] \quad (3.76)$$

where

$$x[n-1] = G\Phi[n-1] + v[n-1]. \quad (3.77)$$

The state variable  $x$  can be updated by the equation derived according to

$$x[n] = \frac{2N}{T}\Phi[n] + v[n] \quad (3.78)$$

$$x[n] = \frac{2N}{T}\Phi[n] + \frac{2N}{T}\Phi[n] - x[n-1] \quad (3.79)$$

$$x[n] = 2\frac{2N}{T}\Phi[n] - x[n-1] \quad (3.80)$$

$$x[n] = 2G\Phi[n] - x[n-1]. \quad (3.81)$$

Similarly, BE discretization leads to the companion derived from

$$v = \frac{Nd\Phi}{dt} \quad (3.82)$$

$$v[n] = \frac{N}{T}(\Phi[n] - \Phi[n-1]) \quad (3.83)$$

$$v[n] = \frac{N}{T}\Phi[n] - x[n-1] \quad (3.84)$$

$$v[n] = G\Phi[n] - x[n-1] \quad (3.85)$$

where

$$x[n-1] = G\Phi[n-1]. \quad (3.86)$$

The state update equation is in this case given by

$$x[n] = G\Phi[n]. \quad (3.87)$$

Both discretization techniques differ in value of conductance  $G = N\alpha$  where  $\alpha = \frac{2}{T}$  for TR and  $\alpha = \frac{1}{T}$  for BE and in the state update equation.

Discretized equations (3.63) – (3.71) with the linear model of transformer core  $\mathcal{F} = \Phi\mathcal{R}$  can be expressed in a matrix form

$$\begin{bmatrix} G_s & 0 & -G_s & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ -G_s & 0 & 2G_s & -G_s & 0 & 0 & 1 & -1 & 1 & 0 & 0 \\ 0 & 0 & -G_s & G_s & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & G_L & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & N_1\alpha \\ 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & N_2\alpha \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & N_3\alpha \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & N_1 & N_2 & N_3 & \mathcal{R} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ i_{ps} \\ i_1 \\ i_2 \\ i_3 \\ \Phi \end{bmatrix} = \begin{bmatrix} i_{s1}(v_1) \\ i_{p1}(v_2) \\ 0 \\ i_{s2}(v_4) \\ i_{p2}(v_5) \\ 0 \\ v_{ps} \\ x_{N_1}[n-1] \\ x_{N_2}[n-1] \\ x_{N_3}[n-1] \\ 0 \end{bmatrix} \quad (3.88)$$

which has the form of MNA matrix equation and which has been already used for the derivation of DK-method matrices. The conductance matrix on the LHS can be divided into a regular part without the transformer model and a part with the connected transformer model. This division is illustrated in Figure 3.12 left. The regular part without the transformer model is denoted by symbol  $\mathbf{S}$  and its obtaining has been described in chapter 1.3.1 e.g. using the equation (1.57). The rest of the matrix contains parameters of the transformer. Several parts can be there further identified, namely connection submatrices which specify which nodes is the transformer connected to, vectors containing numbers of turns of each winding, the transformer core parameter and several zero matrices. Thus, the whole conductance matrix can be built up from submatrices  $\mathbf{S}$ ,  $\mathbf{N}_t$ ,  $\mathbf{N}_w$ , zero matrices and the magnetic reluctance parameter  $\mathcal{R}$  according to Figure 3.12 right.

$$\begin{array}{c|c|c}
 & \begin{array}{c} \xrightarrow{\text{windings}} \\ \text{nodes} \\ \downarrow \end{array} & \mathbf{0} \\
 \mathbf{S} & & \\
 \hline
 \begin{array}{c} \xrightarrow{\text{nodes}} \\ \text{windings} \\ \downarrow \end{array} & \mathbf{0} & \begin{array}{c} N_1\alpha \\ N_2\alpha \\ \vdots \\ N_n\alpha \end{array} \\
 \mathbf{0} & N_1, N_2, \dots, N_n & \mathcal{R}
 \end{array} = \begin{array}{c|c|c}
 \mathbf{S} & \mathbf{N}_t^T & \mathbf{0} \\
 \hline
 \mathbf{N}_t & \mathbf{0} & \mathbf{N}_w\alpha \\
 \hline
 \mathbf{0} & \mathbf{N}_w^T & \mathcal{R}
 \end{array}$$

Figure 3.12: Stamp of linear transformer model for conductance matrix

The matrix  $\mathbf{N}_t$  is the connection matrix with the number of rows given by the number of windings and the number of columns given by the number of circuit nodes increased by the number of inputs. For the circuit from Figure 3.8, it has the form

$$\mathbf{N}_t = \begin{bmatrix} 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}. \quad (3.89)$$

It can be seen that the first winding is connected to nodes 2 and 3, the second winding to nodes 3 and 5 and the third winding to node 6 and the reference node. The matrix  $\mathbf{N}_w$  is in this case

$$\mathbf{N}_w = \begin{bmatrix} N_1 \\ N_2 \\ N_3 \end{bmatrix} \quad (3.90)$$

and generally, the number of rows is given by the number of the windings and it has one column.

There are also three new state variables  $x_{N_1}$ ,  $x_{N_2}$  and  $x_{N_3}$  which must be added to other state variables. To do that, the energy-storing conductance diagonal matrix  $\mathbf{G}_x$  is extended by the vector  $\mathbf{N}_w$  to

$$\mathbf{G}_{xt} = \begin{bmatrix} \mathbf{G}_x & \mathbf{0} \\ \mathbf{0} & \text{diag}(\mathbf{N}_w\alpha) \end{bmatrix}. \quad (3.91)$$

Because state variables  $x_{N_1}$ ,  $x_{N_2}$  and  $x_{N_3}$  act like capacitors according to equations (3.81) and (1.102), these state variables have assigned value 1 in matrix  $\mathbf{Z}$ .

Further, original incidence matrices

$$\mathbf{N}_r = \begin{bmatrix} 1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad (3.92)$$

$$\mathbf{N}_u = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}, \quad (3.93)$$

$$\mathbf{N}_o = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad (3.94)$$

$$\mathbf{N}_n = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}. \quad (3.95)$$

and the currently unused matrix  $\mathbf{N}_v$  must be extended by zero vectors  $\mathbf{0}$  with the number of columns given by number of windings increased by one according to  $\mathbf{N}_{rt} = \begin{bmatrix} \mathbf{N}_r & \mathbf{0} \end{bmatrix}$ ,  $\mathbf{N}_{vt} = \begin{bmatrix} \mathbf{N}_v & \mathbf{0} \end{bmatrix}$ ,  $\mathbf{N}_{ut} = \begin{bmatrix} \mathbf{N}_u & \mathbf{0} \end{bmatrix}$ ,  $\mathbf{N}_{ot} = \begin{bmatrix} \mathbf{N}_o & \mathbf{0} \end{bmatrix}$ ,  $\mathbf{N}_{nt} = \begin{bmatrix} \mathbf{N}_n & \mathbf{0} \end{bmatrix}$  because of the changed size of the conductance matrix. The matrix  $\mathbf{N}_x$ , which is also currently empty for the circuit from Figure 3.8, must be modified to

$$\mathbf{N}_{xt} = \begin{bmatrix} \mathbf{N}_x & \mathbf{0} \\ & 1 & 0 & 0 & 0 \\ \mathbf{0} & 0 & 1 & 0 & 0 \\ & 0 & 0 & 1 & 0 \end{bmatrix} \quad (3.96)$$

which adds the state variables  $x_{N_1}$ ,  $x_{N_2}$  and  $x_{N_3}$  to appropriate positions on the RHS of the equation (1.58). However, we need to express the variable  $\Phi$  from this equation, which is in a different position in the equation (3.88) and therefore we

need the second matrix  $\mathbf{N}_{x2}$  in the form of

$$\mathbf{N}_{xt2} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad (3.97)$$

for the circuit from Figure 3.8 and generally this matrix has the number of rows given by the number of windings and contains 1 values in the last column otherwise zeros. The matrices are obtained from

$$\mathbf{A} = 2\mathbf{Z}\mathbf{G}_{xt}\mathbf{N}_{xt2}\mathbf{S}^{-1}\mathbf{N}_{xt}^T - \mathbf{Z}, \quad (3.98)$$

$$\mathbf{B} = 2\mathbf{Z}\mathbf{G}_{xt}\mathbf{N}_{xt2}\mathbf{S}^{-1} \begin{pmatrix} \mathbf{0} & \mathbf{I} \end{pmatrix}^T, \quad (3.99)$$

$$\mathbf{C} = 2\mathbf{Z}\mathbf{G}_{xt}\mathbf{N}_{xt2}\mathbf{S}^{-1}\mathbf{N}_{nt}^T, \quad (3.100)$$

the other matrices are obtained regularly from equations (1.65) - (1.70) using the new incidence matrices. Finally, the nonlinear function vector is

$$\mathbf{i}(\mathbf{v}) = \begin{bmatrix} -i_s(v_{in1}, v_1) \\ -i_p(v_{in1}, v_1, v_2) \\ -i_s(v_{in2}, v_3) \\ -i_p(v_{in2}, v_3, v_4) \end{bmatrix}. \quad (3.101)$$

Similarly, the incorporation of the nonlinear transformer model into the automated DK-method can be derived. The magnetic reluctance  $\mathcal{R}$  can not be isolated from transformer core models and therefore the circuit equation (3.88) was changed to

$$\begin{bmatrix} G_s & 0 & -G_s & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ -G_s & 0 & 2G_s & -G_s & 0 & 0 & 1 & -1 & 1 & 0 & 0 \\ 0 & 0 & -G_s & G_s & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & G_L & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & N_1\alpha \\ 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & N_2\alpha \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & N_3\alpha \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & N_1 & N_2 & N_3 & 0 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ i_{ps} \\ i_1 \\ i_2 \\ i_3 \\ \Phi \end{bmatrix} = \begin{bmatrix} i_{s1}(v_1) \\ i_{p1}(v_2) \\ 0 \\ i_{s2}(v_4) \\ i_{p2}(v_5) \\ 0 \\ v_{ps} \\ x_{N_1}[n-1] \\ x_{N_2}[n-1] \\ x_{N_3}[n-1] \\ \mathcal{F}(\Phi) \end{bmatrix} \quad (3.102)$$

The conductance matrix  $\mathbf{S}_t$  can be again decomposed into submatrices according to Figure 3.13.

The only change in the conductance matrix is the zero value instead of the magnetic reluctance  $\mathcal{R}$ . Another change is on the RHS where a nonlinear function

$$\begin{array}{c|c|c}
\mathbf{S} & \mathbf{N}_t^T & \mathbf{0} \\
\hline
\mathbf{N}_t & \mathbf{0} & \mathbf{N}_w \alpha \\
\hline
\mathbf{0} & \mathbf{N}_w^T & 0
\end{array}$$

Figure 3.13: Stamp of nonlinear transformer model for the conductance matrix

has been added. Therefore, the incidence vector  $\mathbf{N}_n$  must be modified to

$$\mathbf{N}_{nt} = \begin{bmatrix} \mathbf{N}_n & \mathbf{0} \\ \mathbf{0} & [0 \ 0 \ \dots \ 0 \ 1] \end{bmatrix}. \quad (3.103)$$

One new row was added to introduce one more nonlinear function in the DK-method core function. This vector consists of zero values and 1 at the end of the vector because the equation describing the magnetic circuit is the last one in (3.102).

The last thing to be considered is the DK-method nonlinear core function. The standard form

$$\mathbf{0} = \mathbf{p} + \mathbf{K}\mathbf{i}(\mathbf{v}) - \mathbf{v} \quad (3.104)$$

can be used for the linear and nonlinear transformer with Frohlich model. The nonlinear model function  $i = f(v)$  is equal to (3.7) for Frohlich model where  $i$  corresponds with  $\mathcal{F}$  and  $v$  with  $\Phi$  and the whole nonlinear function vector is then

$$\mathbf{i}(\mathbf{v}) = \begin{bmatrix} -i_s(v_{in1}, v_1) \\ -i_p(v_{in1}, v_1, v_2) \\ -i_s(v_{in2}, v_3) \\ -i_p(v_{in2}, v_3, v_4) \\ -\frac{\bar{c}v_5}{1-\bar{b}|v_5|} \end{bmatrix}. \quad (3.105)$$

This general form is, however, not suitable for the other magnetic core models and the nonlinear core function in the particular form must be introduced. To implement the GC-model with the DK-method, the equation (3.104) is rewritten and GC-model functions are added forming together

$$\begin{aligned}
0 &= p_1 + k_{11}i_1 + k_{12}i_2 + k_{13}i_3 + k_{14}i_4 + k_{15}(i_5 + v_6) - v_1 \\
0 &= p_2 + k_{21}i_1 + k_{22}i_2 + k_{23}i_3 + k_{24}i_4 + k_{25}(i_5 + v_6) - v_2 \\
0 &= p_3 + k_{31}i_1 + k_{32}i_2 + k_{33}i_3 + k_{34}i_4 + k_{35}(i_5 + v_6) - v_3 \\
0 &= p_4 + k_{41}i_1 + k_{42}i_2 + k_{43}i_3 + k_{44}i_4 + k_{45}(i_5 + v_6) - v_4 \\
0 &= p_5 + k_{51}i_1 + k_{52}i_2 + k_{53}i_3 + k_{54}i_4 + k_{55}(i_5 + v_6) - v_5 \\
0 &= \frac{v_5[n] - v_5[n-1]}{T} + i_6
\end{aligned} \quad (3.106)$$



where  $i_1 = -i_s(v_{in1}, v_1)$ ,  $i_2 = -i_p(v_{in1}, v_1, v_2)$ ,  $i_3 = -i_s(v_{in2}, v_3)$ ,  $i_4 = -i_p(v_{in2}, v_3, v_4)$  are functions given by screen and plate current models of the both pentodes. The nonlinear functions  $i_5$  and  $i_6$  come from the GC-model as

$$i_5 = - \left( \frac{v_5}{C} + a \left| \frac{v_5}{C} \right|^n \text{sign}(v_5) \right) \quad (3.107)$$

and

$$i_6 = - \left( \frac{v_6}{r} + b \left| \frac{v_6}{r} \right|^m \text{sign}(v_6) \right) \quad (3.108)$$

where  $v_5 = \Phi$  and  $v_6 = \mathcal{F}_r$ . Because the number of nonlinear equation in (3.106) can differ for different circuit topologies, it can be rewritten into more useful and general form

$$\begin{aligned} \mathbf{0} &= \mathbf{p} + \mathbf{K}\mathbf{i}(\mathbf{v}) - \mathbf{v} + \mathbf{K}_N v_{N+1} \\ 0 &= \frac{v_N[n] - v_N[n-1]}{T} + i_{N+1} \end{aligned} \quad (3.109)$$

where  $N$  is the constant determining number of nonlinear equations and  $\mathbf{K}_N$  is the  $N$ -th column of the  $\mathbf{K}$  matrix and the last nonlinear function  $i_{N+1}$  is not included in the vector  $\mathbf{i}(\mathbf{v})$ .

## 3.2 Simulation of Circuits with Operational Amplifier

Another commonly used circuit component which has not been discussed well in real-time audio effect simulations is the operational amplifier (OPA). The operational amplifier is usually used as the active component of analog filters which can often be found in audio effects circuits. The real-time simulation of analog filters with the operational amplifier was presented in [22]. This work was, however, focused on simple circuits and moreover, the ideal model of the operational amplifier was used. The results showed that it is possible to use the ideal model still with good accuracy. However, there are circuits where the more complex model of the operational amplifier must be used e.g. Low Frequency Oscillator (LFO) circuits.

The operational amplifiers are usually simulated as macro models built up by subcircuits consisting of elementary components modeling the operational amplifier on different levels of complexity [68]. For the case of the real-time audio effect modeling, the ideal, a linear or a behavioral nonlinear model can be used instead of complex OPA macro models. These models can be used because of the limited frequency bandwidth ( $f < 20$  kHz) which allows to use frequency independent models of operational amplifiers. Further, operational amplifiers designed for audio circuits contain a JFET differential input stage which is characterized by the huge input

resistance and thus the input resistance can be often neglected without losing the accuracy of the simulation. The ideal model of an operational amplifier is characterized by infinite amplification. The linear model can be described by

$$v_{\text{out}} = A(v_+ - v_-) \quad (3.110)$$

where  $A$  is the open-load amplification and  $v_+$  is the input voltage at the non-inverting input and  $v_-$  is the input voltage at the inverting input of the operational amplifier.

The nonlinear model of the operational amplifier can be implemented as the behavioral model i.e. by a function which emulates the behavior of the output voltage. A suitable function is the hyperbolic tangens function which can be used for both simulation of limited amplification and output voltage saturation. The nonlinear function can be implemented according to

$$v_{\text{out}} = v_{\text{EE}} + \left( \frac{1}{2} \tanh(a(v_+ - v_-)) + 0.5 \right) (v_{\text{CC}} - v_{\text{EE}}) \quad (3.111)$$

where  $v_{\text{EE}}$  is the negative power supply voltage and  $v_{\text{CC}}$  is the positive power supply voltage and the parameter  $a$  can be used for modeling the different amplification factor  $A$ . The amplification of the model (3.111) is determined by the slope  $f'(x)$  of the function. Therefore, we need to find the first derivative around operating point  $x = v_+ - v_- = 0$ . The first derivative can be derived as

$$\begin{aligned} f'(x)_{x=0} &= 0.5a \left( 1 - \tanh^2(ax) \right) (v_{\text{CC}} - v_{\text{EE}}) \\ &= 0.5a (v_{\text{CC}} - v_{\text{EE}}) \end{aligned} \quad (3.112)$$

which allows isolating the parameter  $a$  as

$$a = \frac{2A}{(v_{\text{CC}} - v_{\text{EE}})} \quad (3.113)$$

where  $A$  is the desired open-load amplification of the model.

The proposed model was tested on two basic circuits – an inverting amplifier and an inverting comparator with schematics given in Figure 3.14.

The nonlinear equation

$$0 = \frac{R_2}{R_1 + R_2} v_{\text{in}} + \frac{R_1}{R_1 + R_2} \left( v_{\text{EE}} + \left( \frac{\tanh(-av)}{2} + 0.5 \right) (v_{\text{CC}} - v_{\text{EE}}) \right) - v \quad (3.114)$$

for unknown variable  $v$  on the inverting OPA input terminal can be derived for the inverting amplifier from Figure 3.14 left. The values of components were  $R_1 = 10 \text{ k}\Omega$ ,  $R_2 = 10 \text{ k}\Omega$ ,  $v_{\text{EE}} = -9 \text{ V}$ ,  $v_{\text{CC}} = 9 \text{ V}$  and  $A = 100000$ . The equation was solved numerically and the resulting transfer function is shown in Figure 3.15.

Similarly, the comparator circuit can be described by the equation

$$0 = \frac{R_1}{R_1 + R_2} \left( v_{\text{EE}} + \left( \frac{\tanh(a(v - v_{\text{in}}))}{2} + 0.5 \right) (v_{\text{CC}} - v_{\text{EE}}) \right) - v \quad (3.115)$$

and its transfer function is shown in Figure 3.16. Both simulated circuits have the proper transfer function, which proves the validity of the proposed model.

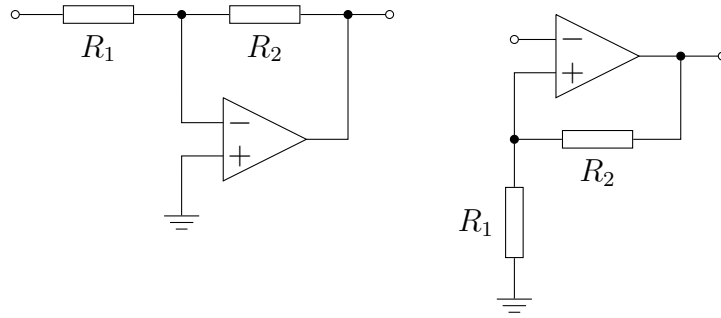


Figure 3.14: Circuit schematic for the inverting amplifier (left) and comparator (right).

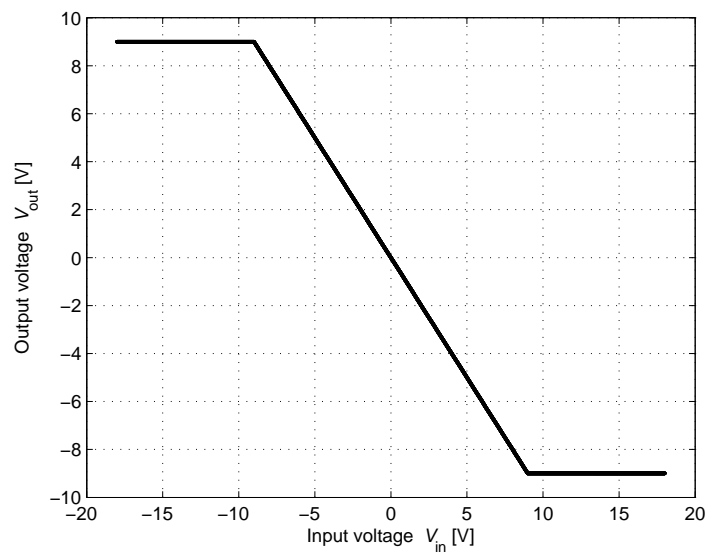


Figure 3.15: Transfer function of the inverting amplifier with model of OPA for real-time processing.

### 3.2.1 Incorporation of Operational Amplifier Model into Automated DK-method

Because the circuits with operational amplifiers are usually much more complex than the circuits in Figures 3.14 and 3.15, the automated derivation of circuit equations would be beneficial. During the process of the derivation of the DK-method, a similar approach, as was used for circuits with the audio transformer, can be used. Firstly, the whole circuit without operational amplifiers can be analyzed using the standard automated DK-method to get the conductance matrix  $\mathbf{S}$  consisting of resistors and discretized models of inductors and capacitors.

Subsequently, the operational amplifier is modeled as a nonlinear voltage con-

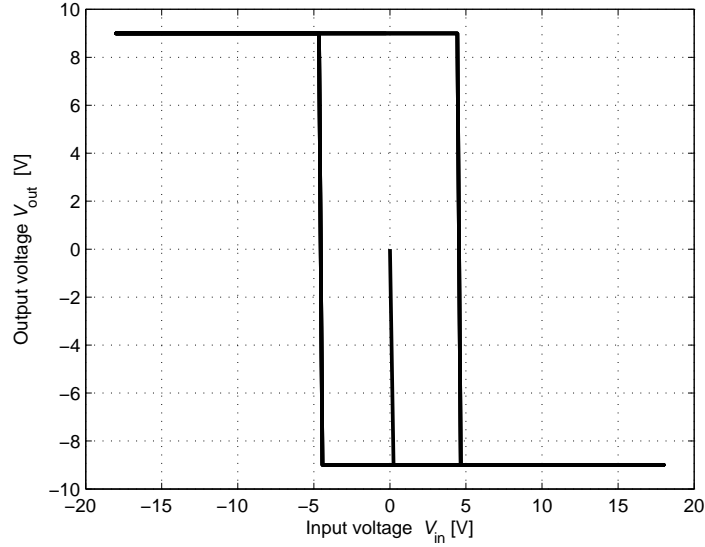


Figure 3.16: Transfer function of the inverting comparator with model of OPA for real-time processing.

trolled voltage source. Therefore, the model of the operational amplifier can be added into the system similarly as any other voltage source resulting in extension of the conductance matrix  $\mathbf{S}$  according to

$$\mathbf{S}_{\text{opa}} = \begin{pmatrix} \mathbf{S} & \mathbf{N}_{\text{opa-O}}^T \\ \mathbf{N}_{\text{opa-O}} & \mathbf{0} \end{pmatrix} \quad (3.116)$$

where the incidence matrix  $\mathbf{N}_{\text{opa-O}}$  holds information where the output of the operational amplifier is connected. The matrix  $\mathbf{N}_{\text{opa-O}}$  can be for one operational amplifier in form

$$\mathbf{N}_{\text{opa-O}} = \begin{bmatrix} \cdots & 0 & 1 & -1 & 0 & \cdots \end{bmatrix} \quad (3.117)$$

where value 1 determines to which node the output is connected and  $-1$  determines from which node the output current is sourced, which is often the reference node and then the  $-1$  element will be missing. The total number of columns of the  $\mathbf{N}_{\text{opa-O}}$  matrix is given by the number of columns of the matrix  $\mathbf{S}$ .

If the circuit operates in such range of voltages that the saturation of the operational amplifier does not occur, the linear model of the operational amplifier given by (3.110) can be used and in this case, the conductance matrix  $\mathbf{S}_{\text{opa}}$  is constructed according to

$$\mathbf{S}_{\text{opa}} = \begin{pmatrix} \mathbf{S} & \mathbf{N}_{\text{opa-O}}^T \\ \mathbf{N}_{\text{opa-O}} + \mathbf{A}\mathbf{N}_{\text{opa-I}} & \mathbf{0} \end{pmatrix} \quad (3.118)$$

where  $\mathbf{A}$  is the diagonal matrix with the amplification factor  $A$  for particular operational amplifiers on the diagonal and  $\mathbf{N}_{\text{opa-I}}$  is the incidence matrix specifying

which nodes the inputs of the operational amplifier are connected to. For one operational amplifier, it can be in form

$$\mathbf{N}_{\text{opa-I}} = \begin{bmatrix} \cdots & 0 & -1 & 1 & 0 & \cdots \end{bmatrix} \quad (3.119)$$

where  $-1$  specifies the inverting input and  $1$  the non-inverting input. If modeling the input resistance of the operational amplifier is desired, the conductance matrix  $\mathbf{S}_{\text{opa}}$  is given by

$$\mathbf{S}_{\text{opa}} = \begin{pmatrix} \mathbf{S} + \mathbf{N}_{\text{opa-I}}^T \mathbf{G}_{\text{opa-in}} \mathbf{N}_{\text{opa-I}} & \mathbf{N}_{\text{opa-O}}^T \\ \mathbf{N}_{\text{opa-O}} + \mathbf{A} \mathbf{N}_{\text{opa-I}} & \mathbf{0} \end{pmatrix} \quad (3.120)$$

where  $\mathbf{G}_{\text{opa-in}}$  is the diagonal matrix with input conductances of particular operational amplifiers on the diagonal.

DK-method matrices are computed using equations (1.62) – (1.70) but first of all, incidence matrices  $\mathbf{N}_r$ ,  $\mathbf{N}_v$ ,  $\mathbf{N}_x$ ,  $\mathbf{N}_o$ ,  $\mathbf{N}_u$  and  $\mathbf{N}_n$  must be extended by zero submatrices similarly as matrices for the transformer model. This approach is valid only for the linear model of the operational amplifier. In the case of the nonlinear model, nonlinear DK-method equations must be extended. Firstly, the control voltage must be obtained. The control voltage is the voltage between the terminals of the operational amplifier and these terminals have already been specified by the matrix  $\mathbf{N}_{\text{opa-I}}$ . However, the construction of the  $\mathbf{K}$  matrix differs from (1.70) because the nonlinear function is not connected to the control voltage nodes but is connected as the additional voltage source. Therefore, the  $\mathbf{K}$  matrix can be obtained from

$$\mathbf{K} = \begin{pmatrix} \mathbf{N}_n & \mathbf{0} \\ \mathbf{N}_{\text{opa-I}} & \mathbf{0} \end{pmatrix} \mathbf{S}_{\text{opa}}^{-1} \left( \begin{pmatrix} \mathbf{N}_n \\ \mathbf{0} \end{pmatrix} + \begin{pmatrix} \mathbf{0} \\ \mathbf{I} \end{pmatrix} \right) \quad (3.121)$$

where the identity matrix is used to express the connection of the nonlinear voltage controlled voltage sources of the operational amplifiers and  $\mathbf{N}_n$  is the incidence matrix of nonlinear components without considering the operational amplifiers. The conductance matrix  $\mathbf{S}_{\text{opa}}$  is computed from (3.116) or from (3.120) where  $\mathbf{A}$  is zero matrix. The matrix  $\mathbf{C}$  is derived using

$$\mathbf{C} = 2\mathbf{G}_x \mathbf{Z} \begin{pmatrix} \mathbf{N}_x & \mathbf{0} \end{pmatrix} \mathbf{S}_{\text{opa}}^{-1} \left( \begin{pmatrix} \mathbf{N}_n \\ \mathbf{0} \end{pmatrix} + \begin{pmatrix} \mathbf{0} \\ \mathbf{I} \end{pmatrix} \right) \quad (3.122)$$

and the matrix  $\mathbf{F}$  from

$$\mathbf{F} = \begin{pmatrix} \mathbf{N}_o & \mathbf{0} \end{pmatrix} \mathbf{S}_{\text{opa}}^{-1} \left( \begin{pmatrix} \mathbf{N}_n \\ \mathbf{0} \end{pmatrix} + \begin{pmatrix} \mathbf{0} \\ \mathbf{I} \end{pmatrix} \right). \quad (3.123)$$

The other matrices are obtained in standard way.

The method was tested on several circuits with operational amplifiers. Simulations of the amplifier and filter circuits provided good results but circuits with the operational amplifier connected as the comparator showed non-realistic behavior although equations are correct. The numerical algorithm found another root of the nonlinear equation that is mathematically valid but does not respect the physical behavior. The situation is illustrated in Figure 3.17. The nonlinear equation  $v = p + Ki(v)$  describing the circuit with the comparator in DK-method form is plotted for LHS =  $v - p$  (blue and red color) and RHS =  $Ki(v)$  (green color) separately. LHS varies with the input voltage signal  $V_{in}$  while the RHS is constant. The solution of the nonlinear equation is given by the intersection of both sides (marked with the circles). One can see that there are up to three roots of the equation for  $V_{in} = 1\text{ V}$  and  $V_{in} = -1\text{ V}$  which are inside the area of the comparator hysteresis. The numerical solver tends to find the root near the zero but physically valid roots are the other ones. Outside the hysteresis loop ( $V_{in} = 6\text{ V}$  and  $V_{in} = -6\text{ V}$ ) there is only one root which can be found by the numerical algorithm without any special requirements on the initial solution of the nonlinear equation. However, it was shown in the previous chapter that it is possible to solve the circuit with the comparator. That is because the equation (3.115) is in different form which can be characterized as  $v = Ki(v + p)$ . This situation is illustrated in Figure 3.18. There are also more roots in the hysteresis area but they are much more suitably situated. Once the saturation (one state of the comparator) is achieved, the position of the root does not change until the opposite input voltage  $V_{in}$ , lying outside the hysteresis loop, is applied. Then, there is only one root of the nonlinear equation representing the opposite state of the comparator. Figure 3.19 shows similar analysis of the circuit with the operational amplifier connected as the amplifier. It can be seen that there is always only one root of nonlinear equations.

As a result, the standard nonlinear core of the DK-method must be rearranged into the form  $v = Ki(v + p)$ , which is however complicated for systems with more nonlinear equations and practically it is suitable only for systems with only one nonlinear equation describing the comparator. For more complex systems, a proper initial solution of the nonlinear equation must be used. Performed simulations showed that it is sufficient to estimate unknown comparator input voltages derived for both states of the comparator, use both of them as the initial solution and finally use the root for which the whole system converges.

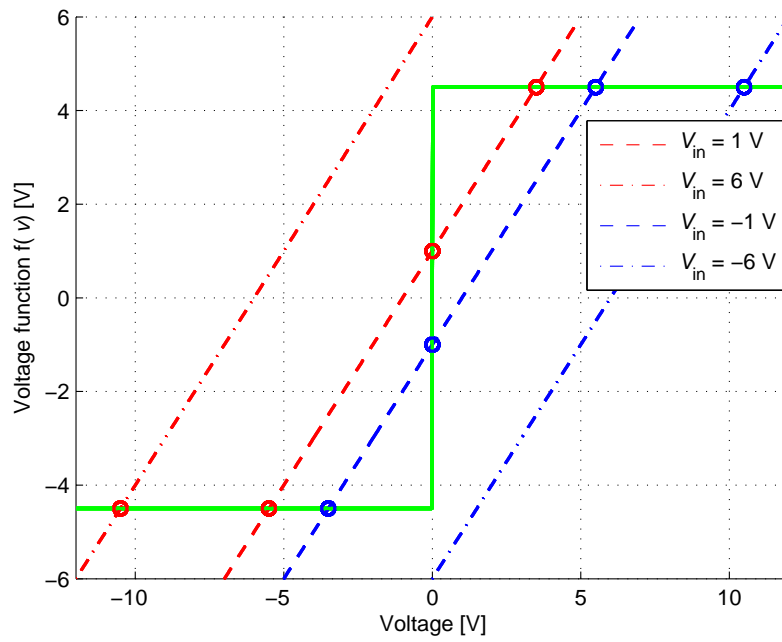


Figure 3.17: Solution of the nonlinear equation for the comparator circuit using DK-method.

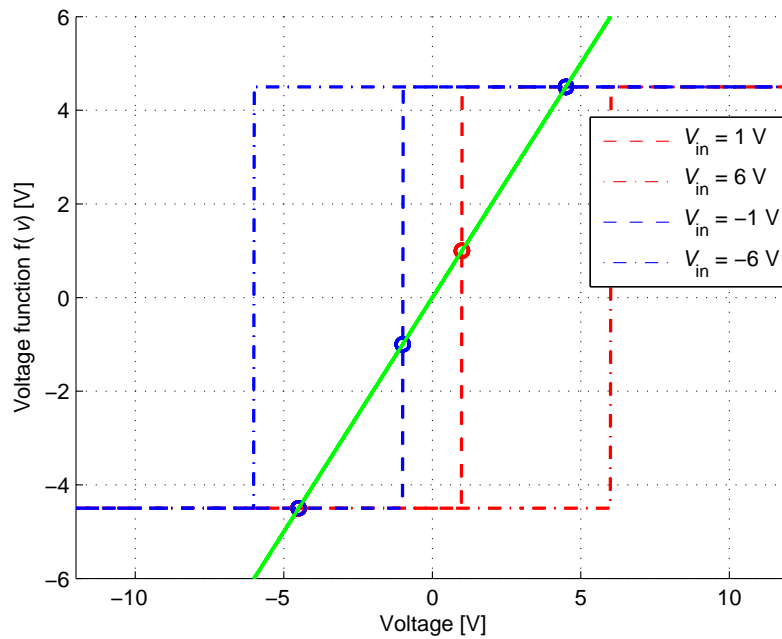


Figure 3.18: Solution of the nonlinear equation for the comparator circuit using equation (3.115).

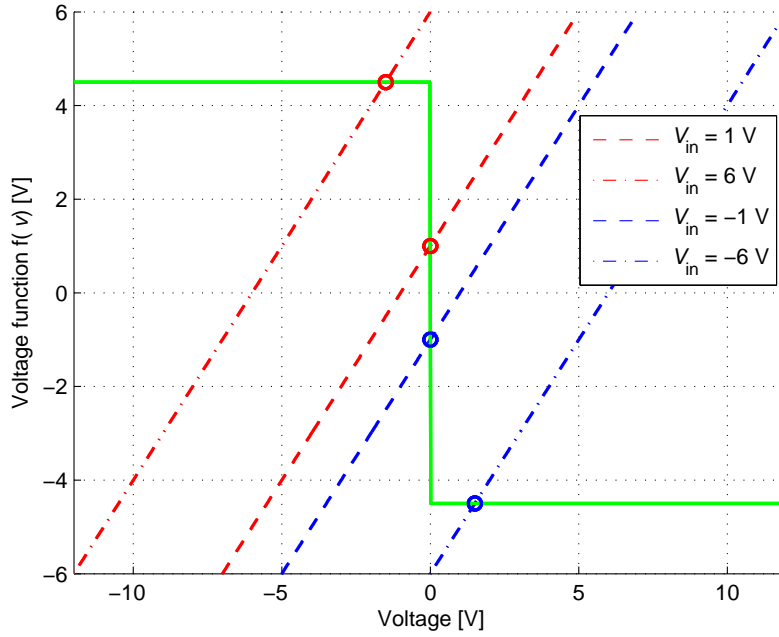


Figure 3.19: Solution of the nonlinear equation for the inverting amplifier.

The DK-method was tested with a simulation of the basic LFO generator consisting of two operational amplifiers. The circuit schematic is shown in Figure 3.20 and values in Table 3.7. The circuit can be described by the matrices

$$\mathbf{N}_r = \begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ -1 & 0 & 0 & 1 \end{bmatrix}, \quad (3.124)$$

$$\mathbf{G}_r = \begin{bmatrix} \frac{1}{R_1} & 0 & 0 \\ 0 & \frac{1}{R_2} & 0 \\ 0 & 0 & \frac{1}{R_3} \end{bmatrix}, \quad (3.125)$$

$$\mathbf{N}_x = \begin{bmatrix} 0 & 1 & -1 & 0 \end{bmatrix}, \quad (3.126)$$

$$\mathbf{G}_x = \begin{bmatrix} C \\ 2T \end{bmatrix}, \quad (3.127)$$

$$\mathbf{N}_{\text{OPA-O}} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}, \quad (3.128)$$

$$\mathbf{N}_{\text{OPA-I}} = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (3.129)$$



All these matrices form according to (3.116) the conductance matrix  $\mathbf{S}_{\text{OPA}}$  given by

$$\mathbf{S}_{\text{OPA}} = \begin{bmatrix} G_1 + G_3 & -G_1 & 0 & G_3 & 0 & 1 \\ -G_1 & G_1 + G_c & -G_c & 0 & 0 & 0 \\ 0 & -G_c & G_c + G_2 & -G_2 & 1 & 0 \\ -G_3 & 0 & -G_2 & G_2 + G_3 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (3.130)$$

and DK-method matrices are computed from equations (3.121), (3.122), (3.123) and (1.62), (1.63), (1.65), (1.66), (1.68) and (1.69) where all the incidence matrices  $\mathbf{N}$  are extended by zero matrix according to the context, incidence matrix  $\mathbf{N}_n$  is the empty matrix and the output matrix is given by

$$\mathbf{N}_o = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}. \quad (3.131)$$

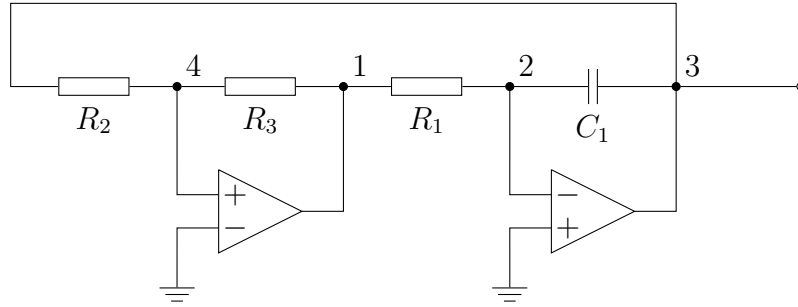


Figure 3.20: Circuit schematic of the simple LFO generator.

Table 3.7: Values of circuit components for the LFO generator.

$R_1$	$R_2$	$R_3$	$C$	$V_{ee}$	$V_{cc}$	$A$
25 k $\Omega$	33 k $\Omega$	47 k $\Omega$	5 $\mu$ F	-4.5 V	4.5 V	$1 \cdot 10^5$

Figure 3.21 shows the transient analysis of the LFO generator circuit. The output signal (blue color) is shown as well as the output signal of the comparator (green color) and the state variable  $X$  (red color). This simulation proved the validity of automated derivation of DK-method matrices and also proved that it is possible to use the DK-method for simulation of self-oscillating circuits.

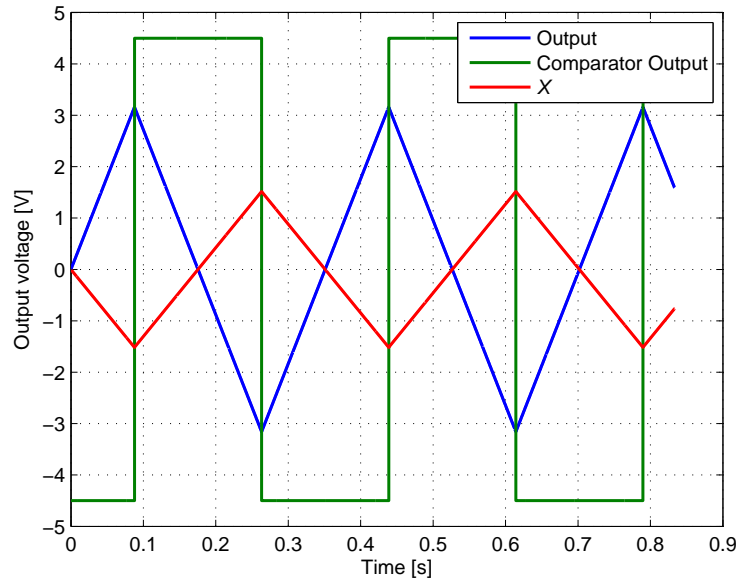


Figure 3.21: Transient analysis of the LFO generator simulated with the DK-method.

### 3.3 Further Considerations Regarding DK-Method

This chapter will address further thoughts which might be useful during designing the simulation algorithm. This covers especially the linearization of the simulated circuit in terms of state-space description and further BE discretization during the automated derivation of DK-method matrices.

The linearization of the circuit can be useful in cases where the input signal is sufficiently small and where, as the result, the produced nonlinear distortion is so low that it is not perceived by human hearing. This allows omitting the nonlinear part of the state-space representation which requires numerical solving and the circuit can be represented in the standard state-space form

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}, \quad (3.132)$$

$$\mathbf{y} = \mathbf{D}\mathbf{x} + \mathbf{E}\mathbf{u} \quad (3.133)$$

with slightly different notation of matrices ( $\mathbf{D}$  instead of  $\mathbf{C}$  and  $\mathbf{E}$  instead of  $\mathbf{D}$ ) used to respect the nonlinear state-space representation notation. The state-space model can be of course constructed from linear small signal models of the nonlinear circuit component but parameters strongly depend on the operating point position. Therefore, the nonlinear model of the circuit must be constructed in order to find the nonlinear DC solution and operating points anyway. And once the nonlinear model has been constructed, it is needless to build the new linear circuit model from the beginning and it is easier to simplify the nonlinear model.

Considering the nodal analysis, small signal models of nonlinear circuit components are represented with admittance parameters –  $y$  parameters which are derived as

$$y = \left. \frac{\partial i}{\partial v} \right|_{v_{\text{DC}}} \quad (3.134)$$

in operating point  $v_{\text{DC}}$  of the device for one port component. A two port component is described by four  $y$  parameters

$$\begin{bmatrix} y_{11} & y_{12} \\ y_{21} & y_{22} \end{bmatrix} = \begin{bmatrix} \left. \frac{\partial i_1}{\partial v_1} \right|_{v_{\text{DC}}} & \left. \frac{\partial i_1}{\partial v_2} \right|_{v_{\text{DC}}} \\ \left. \frac{\partial i_2}{\partial v_1} \right|_{v_{\text{DC}}} & \left. \frac{\partial i_2}{\partial v_2} \right|_{v_{\text{DC}}} \end{bmatrix} = \mathbf{J}(v_1, v_2) \quad (3.135)$$

where  $\mathbf{J}(v_1, v_2)$  is the Jacobian matrix of nonlinear functions  $i_1 = f(v_1, v_2)$ ,  $i_2 = f(v_1, v_2)$  determining the nonlinear model of the given device. Similarly, the whole nonlinear part of the circuit can be considered as a multi-port nonlinear device given by functions  $\mathbf{i} = f(\mathbf{v})$  whose  $y$  parameters are determined by the Jacobian matrix in the operating point. Subsequently, the equation (1.56) can be modified into

$$\mathbf{S} \begin{pmatrix} \mathbf{V} \\ \mathbf{i}_s \end{pmatrix} = \begin{pmatrix} \mathbf{N}_x^T \\ \mathbf{0} \end{pmatrix} \mathbf{x} + \begin{pmatrix} \mathbf{0} \\ \mathbf{I} \end{pmatrix} \mathbf{u} - \begin{pmatrix} \mathbf{N}_n^T \\ \mathbf{0} \end{pmatrix} \mathbf{J}_{v_{\text{DC}}} \begin{pmatrix} \mathbf{N}_n & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{V} \\ \mathbf{i}_s \end{pmatrix} \quad (3.136)$$

where  $\mathbf{J}_{v_{\text{DC}}}$  is the Jacobian matrix of the whole system. The linearized part can be moved to LHS to get

$$\left( \mathbf{S} + \begin{pmatrix} \mathbf{N}_n^T \\ \mathbf{0} \end{pmatrix} \mathbf{J}_{v_{\text{DC}}} \begin{pmatrix} \mathbf{N}_n & \mathbf{0} \end{pmatrix} \right) \begin{pmatrix} \mathbf{V} \\ \mathbf{i}_s \end{pmatrix} = \begin{pmatrix} \mathbf{N}_x^T \\ \mathbf{0} \end{pmatrix} \mathbf{x} + \begin{pmatrix} \mathbf{0} \\ \mathbf{I} \end{pmatrix} \mathbf{u}. \quad (3.137)$$

The voltages across energy storing components can be expressed

$$\mathbf{V}_x = \begin{pmatrix} \mathbf{N}_x & \mathbf{0} \end{pmatrix} \left( \mathbf{S} + \begin{pmatrix} \mathbf{N}_n^T \\ \mathbf{0} \end{pmatrix} \mathbf{J}_{v_{\text{DC}}} \begin{pmatrix} \mathbf{N}_n & \mathbf{0} \end{pmatrix} \right)^{-1} \left( \begin{pmatrix} \mathbf{N}_x^T \\ \mathbf{0} \end{pmatrix} \mathbf{x} + \begin{pmatrix} \mathbf{0} \\ \mathbf{I} \end{pmatrix} \mathbf{u} \right) \quad (3.138)$$

and similarly output voltages are given by

$$\mathbf{V}_o = \begin{pmatrix} \mathbf{N}_o & \mathbf{0} \end{pmatrix} \left( \mathbf{S} + \begin{pmatrix} \mathbf{N}_n^T \\ \mathbf{0} \end{pmatrix} \mathbf{J}_{v_{\text{DC}}} \begin{pmatrix} \mathbf{N}_n & \mathbf{0} \end{pmatrix} \right)^{-1} \left( \begin{pmatrix} \mathbf{N}_x^T \\ \mathbf{0} \end{pmatrix} \mathbf{x} + \begin{pmatrix} \mathbf{0} \\ \mathbf{I} \end{pmatrix} \mathbf{u} \right). \quad (3.139)$$

Subsequently, state-space matrices can be computed from

$$\mathbf{A} = 2\mathbf{Z}\mathbf{G}_x \begin{pmatrix} \mathbf{N}_x & \mathbf{0} \end{pmatrix} \mathbf{S}_{\text{Lin}}^{-1} \begin{pmatrix} \mathbf{N}_x & \mathbf{0} \end{pmatrix}^T - \mathbf{Z}, \quad (3.140)$$

$$\mathbf{B} = 2\mathbf{Z}\mathbf{G}_x \begin{pmatrix} \mathbf{N}_x & \mathbf{0} \end{pmatrix} \mathbf{S}_{\text{Lin}}^{-1} \begin{pmatrix} \mathbf{0} & \mathbf{I} \end{pmatrix}^T, \quad (3.141)$$

$$\mathbf{D} = \begin{pmatrix} \mathbf{N}_o & \mathbf{0} \end{pmatrix} \mathbf{S}_{\text{Lin}}^{-1} \begin{pmatrix} \mathbf{N}_x & \mathbf{0} \end{pmatrix}^T, \quad (3.142)$$

$$\mathbf{E} = \begin{pmatrix} \mathbf{N}_o & \mathbf{0} \end{pmatrix} \mathbf{S}_{\text{Lin}}^{-1} \begin{pmatrix} \mathbf{0} & \mathbf{I} \end{pmatrix}^T \quad (3.143)$$

where

$$\mathbf{S}_{\text{Lin}} = \left( \mathbf{S} + \begin{pmatrix} \mathbf{N}_n^T \\ \mathbf{0} \end{pmatrix} \mathbf{J}_{\mathbf{v}_{\text{DC}}} \begin{pmatrix} \mathbf{N}_n & \mathbf{0} \end{pmatrix} \right) \quad (3.144)$$

and other matrices are equal to matrices used for the nonlinear state-space representation. Note that the linearized model does not include DC offset signal vector. This is because linearizes small signal models usually do not include DC offsets. The only missing matrix is the Jacobian matrix  $\mathbf{J}_{\mathbf{v}_{\text{DC}}}$  defined in the operating point. The operating point can be determined from DC solution of the nonlinear state-space model. The direct computation of the unknown voltages from the equation (1.28) is not possible due to the unknown state vector  $\mathbf{x}$ . However,  $\mathbf{x}$  does not change with the constant input vector  $\mathbf{u}$  and therefore the state vector  $\mathbf{x}$  can be expressed from the state update equation

$$\begin{aligned} \mathbf{x} &= \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} + \mathbf{C}\mathbf{i} \Rightarrow \\ \mathbf{x} &= (\mathbf{I} - \mathbf{A})^{-1} (\mathbf{B}\mathbf{u} + \mathbf{C}\mathbf{i}) \end{aligned} \quad (3.145)$$

and it can be substituted into the nonlinear equation (1.28) and rearranged according to

$$\begin{aligned} \mathbf{0} &= \mathbf{G}\mathbf{x} + \mathbf{H}\mathbf{u} + \mathbf{K}\mathbf{i}(\mathbf{v}_{\text{DC}}) - \mathbf{v}_{\text{DC}} \Rightarrow \\ \mathbf{0} &= \mathbf{G}(\mathbf{I} - \mathbf{A})^{-1} (\mathbf{B}\mathbf{u} + \mathbf{C}\mathbf{i}) + \mathbf{H}\mathbf{u} + \mathbf{K}\mathbf{i}(\mathbf{v}_{\text{DC}}) - \mathbf{v}_{\text{DC}} \Rightarrow \\ \mathbf{0} &= \left( \mathbf{G}(\mathbf{I} - \mathbf{A})^{-1} \mathbf{B} + \mathbf{H} \right) \mathbf{u} + \left( \mathbf{G}(\mathbf{I} - \mathbf{A})^{-1} \mathbf{C} + \mathbf{K} \right) \mathbf{i}(\mathbf{v}_{\text{DC}}) - \mathbf{v}_{\text{DC}} \end{aligned} \quad (3.146)$$

and using this equation it is possible to compute DC voltages  $\mathbf{v}_{\text{DC}}$  without knowing the state vector  $\mathbf{x}$  and subsequently compute the Jacobian matrix  $\mathbf{J}_{\mathbf{v}_{\text{DC}}}$ .

The computational cost of the linearized system can be further reduced by converting the state-space representation into the transfer function  $H(z)$  according to the formula [90]

$$H(z) = \mathbf{D}(z\mathbf{I} - \mathbf{A})^{-1} \mathbf{B} + \mathbf{E} \quad (3.147)$$

and subsequently, the whole system can be implemented as a standard digital filter which provides very efficient implementation.

The TR discretization was considered during the automated derivation of DK-method matrices so far. However, there might be situations where the BE discretization is desired. This foremost affects discretized inductors and capacitors. The current flowing through these devices continues fulfilling the equation (1.101), only the value of conductances have to be changed to  $G_x = \frac{C}{T}$  for the capacitor and  $G_x = \frac{T}{L}$  for the inductor. The state update equation for the capacitor

$$x[n] = G_x v[n] \quad (3.148)$$

while the state update equation for the inductor remains in the same form

$$x[n] = -G_x v[n] + x[n-1] \quad (3.149)$$

as with TR discretization. Both equations can be unified by

$$x[n] = ZG_x v[n] + \left(\frac{Z-1}{2}\right) x[n-1] \quad (3.150)$$

where  $Z$  has the same value as in (1.102). If we introduce a parameter  $m$  equal to  $m = 1$  for BE discretization and  $m = 2$  for TR discretization, equations (1.102) and (3.150) can be unified to

$$x[n] = mZG_x v[n] + \left(\frac{Z-2+m}{3-m}\right) x[n-1]. \quad (3.151)$$

This change affects matrices  $\mathbf{A}$ ,  $\mathbf{B}$  and  $\mathbf{C}$ , which are further given as

$$\mathbf{A} = m\mathbf{Z}\mathbf{G}_x \begin{pmatrix} \mathbf{N}_x & \mathbf{0} \end{pmatrix} \mathbf{S}^{-1} \begin{pmatrix} \mathbf{N}_x & \mathbf{0} \end{pmatrix}^T - \left(\frac{\mathbf{Z} - \mathbf{I}(2+m)}{3-m}\right), \quad (3.152)$$

$$\mathbf{B} = m\mathbf{Z}\mathbf{G}_x \begin{pmatrix} \mathbf{N}_x & \mathbf{0} \end{pmatrix} \mathbf{S}^{-1} \begin{pmatrix} \mathbf{0} & \mathbf{I} \end{pmatrix}^T, \quad (3.153)$$

$$\mathbf{C} = m\mathbf{Z}\mathbf{G}_x \begin{pmatrix} \mathbf{N}_x & \mathbf{0} \end{pmatrix} \mathbf{S}^{-1} \begin{pmatrix} \mathbf{N}_n & \mathbf{0} \end{pmatrix}^T \quad (3.154)$$

and similarly matrices for the linearized system.

The last remark concerns the sign of nonlinear device models  $i = f(v)$ . The automated DK-method defined by equations (1.62) – (1.70) requires the opposite sign of the device models defined in the chapter 1.4 as it was shown e.g. in the equation (3.105). This is because nonlinear currents act as current sources in equation (1.56) and the positive sign means in terms of MNA that the current flows into a node. However, in reality, these nonlinear currents flow out of the node and therefore, nonlinear device models must be used with the opposite sign or the equation (1.56) should be in form

$$\mathbf{S} \begin{pmatrix} \mathbf{V} \\ \mathbf{i}_s \end{pmatrix} = \begin{pmatrix} \mathbf{N}_x^T \\ \mathbf{0} \end{pmatrix} \mathbf{x} + \begin{pmatrix} \mathbf{0} \\ \mathbf{I} \end{pmatrix} \mathbf{u} - \begin{pmatrix} \mathbf{N}_n^T \\ \mathbf{0} \end{pmatrix} \mathbf{i}_n \quad (3.155)$$

which leads to defining the matrices  $\mathbf{C}$ ,  $\mathbf{F}$  and  $\mathbf{K}$  as

$$\mathbf{C} = -m\mathbf{Z}\mathbf{G}_x \begin{pmatrix} \mathbf{N}_x & \mathbf{0} \end{pmatrix} \mathbf{S}^{-1} \begin{pmatrix} \mathbf{N}_n & \mathbf{0} \end{pmatrix}^T, \quad (3.156)$$

$$\mathbf{F} = -\begin{pmatrix} \mathbf{N}_o & \mathbf{0} \end{pmatrix} \mathbf{S}^{-1} \begin{pmatrix} \mathbf{N}_n & \mathbf{0} \end{pmatrix}^T, \quad (3.158)$$

$$\mathbf{K} = -\begin{pmatrix} \mathbf{N}_n & \mathbf{0} \end{pmatrix} \mathbf{S}^{-1} \begin{pmatrix} \mathbf{N}_n & \mathbf{0} \end{pmatrix}^T. \quad (3.159)$$

However, both approaches are possible and valid.

## 3.4 Summary

This chapter adopted the automated DK-method based on incidence matrices of circuit components introduced by Holters and Zölzer in [66]. The contribution is the extension of this approach by incorporating more complicated circuit component models. The first circuit component is the model of the audio transformer. The ideal, linear and nonlinear model of the transformer was discussed. The nonlinear model of the transformer requires use of the nonlinear magnetic core model. Three models of the magnetic core were tested on two circuits – the input stage with the transformer and the push-pull amplifier. The Jiles-Atherton model provides very good simulation of the magnetic core but it is not very suitable for real-time applications. The Frohlich model and the GC model are more suitable for these purposes. The GC model provides modeling of the hysteresis when compared to the Frohlich model. However, as it was shown, the hysteresis manifests very slightly in the output signal. The GC model however also provides better modeling of the shape of the magnetization curve. Therefore, it was chosen as the model used in further simulations. Finally, the method of automated incorporating of the transformer model into the DK-method was introduced, which can facilitate design of real-time simulation algorithms

The second part dealt with the incorporation of the operational amplifier into the DK-method. First of all, the memoryless nonlinear model of the operational amplifier was designed. Secondly, this model was tested on two basic circuits – the amplifier and the comparator. This model was also successfully integrated into the DK method. In addition to the saturation and the finite amplification of the operational amplifier, it is also possible to consider the input resistance of the operational amplifier. The output resistance is not considered in this model, but it can be easily simulated by connecting a resistor in series with the output port of the operational amplifier. The validity of incorporation of the operational amplifier model into the DK-method was proved on simulation of the simple LFO generator.

The last part discussed foremost the linearization of the circuit using the designed nonlinear model of the circuit. This enables reduction of the computational complexity if the circuit operates almost in the linear area. This reduction is not very high because the nonlinear system converges usually in one iteration in these cases. However, when the system is linearized, it is possible to convert the state-space representation into the transfer function representation and implement it as a standard digital filter which can offer more efficient implementation. This linearization can be useful when designing the simulation of audio effects based on active filters (equalizer, phaser effect, etc.)

## 4 APPROXIMATION OF IMPLICIT NONLINEAR CIRCUIT EQUATIONS

Most of modern algorithms for the simulation of nonlinear audio systems require numerical algorithms to compute the output signal. This is due to the implicit form of nonlinear equations which describe the simulated system

$$\mathbf{y} = \mathbf{f}(\mathbf{y}, \mathbf{u}, \dots). \quad (4.1)$$

It is sometimes possible to find an analytical solution of 4.1, however it depends on nonlinear function  $\mathbf{f}(\dots)$  complexity and in general, the solution must be transferred into a nonlinear function root finding problem. The direct usage of numerical algorithms is however not very convenient for real-time applications. It is foremost due to two properties – high computational complexity and variable computational complexity, which does not allow the efficient utilization of the DSP system computational power. Moreover, the numerical solution can also diverge if suitable initial conditions are not used or the solution can converge to local minimum or to another root (as was shown in the previous chapter).

Therefore, the usage of approximation of the function (4.1) is essential for real-time applications. It ensures the constant computational complexity and unique solutions. The computational complexity is also lower than the computational complexity of numerical algorithms but it depends on particular approximation algorithm. The drawback is necessity of look-up tables with stored function values. The look-up tables can require large amount of data stored in memory of the DSP system especially when a multi-dimensional function is approximated. The choice of the approximation algorithm is result of trade-off between the computational complexity, accuracy of approximation and size of the look-up tables and thus, various approximation algorithms will be examined in this chapter.

### 4.1 Precomputation of Nonlinear Systems

The precomputation of the nonlinearity is the first step of the approximation of given nonlinearity. The term precomputation means obtaining function values of a nonlinear equation or set of nonlinear equations in form of (4.1) where  $\mathbf{y}$  is the vector of unknown variables for any input values given by the input vector  $\mathbf{u}$ . This process requires numerical solution of the nonlinear equation using the multi-dimensional Newton method for each element of input vector, and thus this process is computational very demanding. Moreover, boundary values of the input vector can hold large values and the numerical solution does not have to converge to

the right solution if sufficiently close initial values are not known. To cope with this problem, the initial estimation can be derived from the solution for neighbouring values of the input vector  $\mathbf{u}$ . The estimation can be sufficiently accurate if there is small step between the neighbouring input values. It is also practical to start the precomputation from zero input values and continue with positive and negative input values separately up to the boundary values of input vector. Further, Yeh in his work [22] suggested to use the homotopy to aid convergence. The range of the input vector can be determined from input signal properties but unfortunately, the step of the input values has to be chosen empirically.

While the dimension of the nonlinearity depends on the number of circuit nodes or nonlinear equations, the dimension of the approximated function depends on the number of inputs of the nonlinearity. The approximation of the nonlinearity is not limited only for the certain method of simulation (e.g. MNA method, WDF or K-method), but the approximating function can have different dimension for each of those methods.

#### 4.1.1 Precomputation for approximation of nonlinear ODEs

The number of unknowns of the MNA method to be solved is equal or higher than the number of circuit nodes, which is more than the number of nonlinear functions in the given circuit. But the dimension of approximation depends on the number of inputs and states of energy-storing components in this case. For some circuits, the number of inputs is lower than the number of nonlinear equations and therefore the direct approximation of the equations requires lower dimension and also lower computational cost than if the DK-method is used for the simulation. The direct approximation is convenient e.g. for the simulation of blocks used with the modified block-wise simulation of a guitar preamp, which will be discussed in chapter 5.1.

Nonlinear equations to be solved are in form of

$$\mathbf{0} = \mathbf{f}(v_{\text{in}}, v_{x1}, v_{x2}, \dots, v_{xM}) \quad (4.2)$$

where  $v_{\text{in}}$  is the input voltage or the input signal value and  $v_{x1}, v_{x2}, \dots, v_{xM}$  are voltages on energy-storing components. The system has  $N = M + 1$  inputs and a particular solution for a combination of inputs. As a result of precomputation, voltages on all nodes are known, but not all of them are needed. Only the output signal node voltage and new voltages on energy-storing components will be used and they have to be approximated. The system thus have  $N$  outputs and if the functions to be approximated are denoted as  $f_{\text{out}}, f_{x1}, f_{x2}, \dots, f_{xM}$ , the whole system may be



rewritten into

$$\begin{aligned}
v_{\text{out}}[n] &= f_{\text{out}}(v_{\text{in}}[n], v_{x1}[n], v_{x2}[n], \dots, v_{xM}[n]) \\
v_{x1}[n+1] &= v_{x1}[n] + T f_{x1}(v_{\text{in}}[n], v_{x1}[n], v_{x2}[n], \dots, v_{xM}[n]) \\
v_{x2}[n+1] &= v_{x2}[n] + T f_{x2}(v_{\text{in}}[n], v_{x1}[n], v_{x2}[n], \dots, v_{xM}[n]) \\
&\vdots \\
v_{xM}[n+1] &= v_{xM}[n] + T f_{xM}(v_{\text{in}}[n], v_{x1}[n], v_{x2}[n], \dots, v_{xM}[n])
\end{aligned} \tag{4.3}$$

where  $T$  is the sample period, and  $n$  denotes the time index. It would be of course possible to approximate the new voltages on the energy-storing components directly, instead of  $v_{x1}[n+1] = v_{x1}[n] + T f_{x1}(v_{\text{in}}[n], v_{x1}[n], v_{x2}[n], \dots, v_{xM}[n])$ , but this form enables independence of the approximating functions on the sampling frequency.

The construction of the functions  $f_{\text{out}}, f_{x1}, f_{x2}, \dots, f_{xM}$  will be shown on the simple example given by the circuit schematic in Figure 4.1. The circuit can be described by equations

$$\begin{aligned}
0 &= G_2 \frac{v_{\text{in}} G_1 + v_g G_2}{G_1 + G_g + G_2} - v_g G_2 - i_g \\
0 &= v_{c1} - v_k - \frac{v_k G_k - i_p - i_g}{C_k f_s} \\
0 &= v_{\text{ps}} G_p - v_p G_p - V_p G_L - i_p
\end{aligned} \tag{4.4}$$

with unknown variables  $v_g, v_k, v_p$ , the capacitor voltage  $v_{c1}$  from the last sample period, the input signal voltage  $v_{\text{in}}$  and triode nonlinear currents  $i_g$  and  $i_p$ . This means that the system contains two inputs  $v_{c1}$  and  $v_{\text{in}}$ .

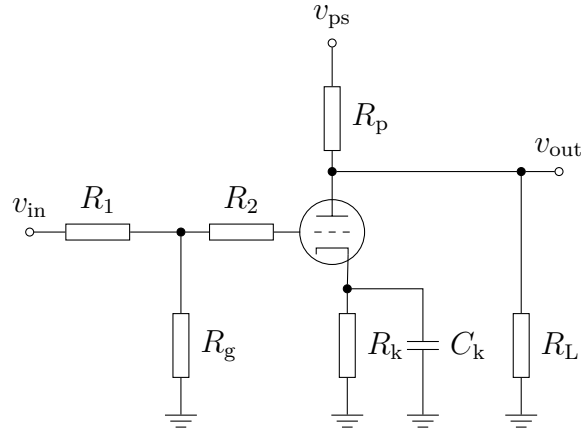


Figure 4.1: Circuit schematic for triode tube amplifier.

After the precomputation for combinations of inputs, all variables  $v_g, v_k, v_p$  are known for the given input. The output function to be approximated is directly the plate voltage  $v_p$

$$f_{\text{out}}(v_{\text{in}}, v_{c1}) = v_p(v_{\text{in}}, v_{c1}).$$

The state update function  $f_{c1}(v_{in}, v_{c1})$  can be derived from the discretized capacitor model

$$\begin{aligned} v_{c1}[n+1] &= v_{c1}[n] + T \frac{v_k G_k - i_p(v_{gk}, v_{pk}) - i_g(v_{gk}, v_{pk})}{C_k} \\ &= v_{cm}[n] + T f_{c1}(v_{in}, v_{c1}) \end{aligned} \quad (4.5)$$

where  $v_{gk} = v_g - v_k$  and  $v_{pk} = v_p - v_k$ . After the approximation of functions  $f_{out}(v_{in}, v_{c1})$  and  $f_{c1}(v_{in}, v_{c1})$ , the real-time simulation only requires the computation of

$$\begin{aligned} v_{out}[n] &= f_{out}(v_{in}[n], v_{c1}[n-1]), \\ v_{c1}[n] &= v_{c1}[n-1] + T f_{c1}(v_{in}[n], v_{c1}[n-1]) \end{aligned}$$

for each input sample.

The computational complexity depends on the number of circuit inputs  $N$  (input signal and states). The computation of one output signal sample requires  $M$  add operations,  $M$  multiply operations and  $N$  computations of approximating functions. Therefore, the total computational complexity markedly depends on the chosen approximation.

### 4.1.2 Precomputation for approximation of the state-space nonlinearity

The core of the state-space nonlinearity (DK-method) is given by the implicit formulation

$$\mathbf{v}[n] = \mathbf{K}\mathbf{i}[n] + \mathbf{p}[n] = \mathbf{K}f(\mathbf{v}[n]) + \mathbf{p}[n] \quad (4.6)$$

for unknown variables  $\mathbf{v}$ . We look for approximating functions in form  $\mathbf{i} = \mathbf{f}(\mathbf{p})$ . Considering the precomputation of such system, the maximal dimension of the precomputed solution will be given by the number of inputs  $\mathbf{p}$ , however some input parameters  $\mathbf{p}$  can be constant for some circuit topologies and therefore the dimension can be lower than the maximal and it is equal to the number of non-constant inputs. For parametric circuits, one has to also consider that the  $\mathbf{K}$  coefficient matrix is dependent on parameter values and then the dimension of the precomputed solution must be extended by the number of variable  $\mathbf{K}$  coefficients, which leads to the maximal dimension given by  $N^2 + N$  where  $N$  is the number of nonlinear functions. The solution can also be precomputed for variable circuit parameters as input variables to the system – this is more efficient when the number of circuit parameters is lower than the number of  $\mathbf{K}$  coefficients which are being changed

by these parameters. As a result in both cases, interpolation functions of high dimensions must be used even for quite simple circuits.

Firstly, let us consider only circuits with constant circuit components. The nonlinearity is being precomputed for inputs  $\mathbf{p}$ . The ranges can be determined from  $\mathbf{p} = \mathbf{G}\mathbf{x} + \mathbf{H}\mathbf{u}$  for the combination of the minimal and maximal values of input  $\mathbf{u}$  and state  $\mathbf{x}$  variables. The step of the input vector values has to be chosen again empirically.

If the number of variable inputs (this excludes constant power supply voltage inputs) and states is lower than the number of inputs  $\mathbf{p}$ , which is equal to the number of nonlinear circuit currents, the dimension of approximating functions is higher than it necessarily has to be and the approximation is then redundant. Some inputs  $\mathbf{p}$  can be constant or linearly dependent and this dependence can be found out from the examination of matrices  $\mathbf{G}$  and  $\mathbf{H}$ . Inputs and states can be concatenated into one input vector and similarly matrices  $\mathbf{G}$  and  $\mathbf{H}$ . The inputs  $\mathbf{p}$  are then given by

$$\mathbf{p} = \begin{bmatrix} \mathbf{G} & \mathbf{H} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{u} \end{bmatrix} \quad (4.7)$$

The equation can be further supplemented with the information about constant inputs

$$\mathbf{p} = \begin{bmatrix} \mathbf{G} & \mathbf{H} \end{bmatrix} \mathbf{u}_{\text{const}} \begin{bmatrix} \mathbf{x} \\ \mathbf{u} \end{bmatrix} \quad (4.8)$$

where  $\mathbf{U}_{\text{const}}$  is the diagonal matrix with ones for variable inputs and zeros for constant inputs on the main diagonal. Linearly independent and non-constant inputs can be found using the Singular Value Decomposition (SVD). The number of variable inputs is given by the number of non-zero singular values of the matrix  $\mathbf{P} = \begin{bmatrix} \mathbf{G} & \mathbf{H} \end{bmatrix} \mathbf{u}_{\text{const}}$  and positions of linearly dependent or constant inputs  $\mathbf{p}$  are given by zero rows of the matrix  $\mathbf{\Sigma}$  of SVD. This analysis is useful for the simulation of more complex systems where the dependence of inputs is not obvious at the first sight.

Parametric circuits are characteristic with variable  $\mathbf{K}$  matrix coefficients and therefore the  $\mathbf{K}$  matrix coefficients or directly values of parametric circuit components are additional inputs of approximating functions and thus the precomputation must include the numerical solution for these additional inputs. Ranges of  $\mathbf{K}$  matrix coefficients are computed from the DK-analysis of the circuit for the minimal and maximal values of given variable parameters or they are directly given by the minimal and maximal values of parameters. But parameter values are never given very precisely, especially in case of analog circuits with potentiometers and therefore parameter input values do not have to be sampled as with small step as the input parameters  $\mathbf{p}$  and often it is sufficient to precompute the nonlinear function only for the boundary values of parameters.

## 4.2 Brief Overview of Function Approximation Techniques

This chapter gives a brief overview of several algorithms for function approximations. Discussed approximation algorithms can be found in literature [91, 92, 93]. The algorithms can be generally divided into least square error approximation algorithms and interpolation algorithms. The least square error approximation minimizes the overall error but the approximating function does not have to go through the function values. The polynomial least square error approximation with polynomial coefficients computed from the Vandermonde matrix is the most suitable algorithm from this group. However, as it has been already mentioned in chapter 1.2.3, the approximation of transfer functions in wider range of input values requires high order polynomials which are inefficient to compute. For these cases, much more suitable approach is to use an interpolation between tabulated function values. It is also possible to construct an interpolating polynomial going through tabulated function values. This polynomial can be e.g. the Lagrange polynomial. This approximation is however computationally very inefficient until the Neville's algorithm is used for constructing the Lagrange polynomial. But this approximation also leads to the high order polynomial when the nonlinear transfer function is approximated in the wider range and moreover, the approximating polynomial can be bad-behaved between the tabulated values.

The most suitable algorithms are therefore algorithms based on the piece-wise interpolation e.g. the linear interpolation or the spline interpolation. From the quality of the approximation point of view, individual algorithms differ foremost in smoothness of the overall approximating function. The smoothness of the approximating function is related to  $r$ -derivatives continuity and can be categorized by the  $C^r$  class.

Because the aim is also to determine computational complexity of several approximation algorithms, formulas as well as brief comparison is given in Appendix A.

## 4.3 Implementation and comparison of approximation of 1-D function

The efficient implementation of approximating functions is very important task when implementing the simulation algorithm in real-time. This chapter is focused on the comparison of algorithms reviewed in the previous chapter. All of these algorithms are based on the interpolation between tabulated function values. The key aspect of

the efficient implementation is, together with the implementation of the interpolating algorithm, efficient looking up of function values used for interpolation. There are plenty of approaches for looking up the values. They are usually based on the linear or binary searching of the table key given by input value  $x$  in the sorted look-up table or array. In this case, the input key value  $x$  is the interpolation break point  $x_i$ . Both these methods do not provide the constant computational cost of the searching but the computational cost depends on the size of the look-up table, which can be quite large. However, it is also possible to derive the index of the searched function value  $f(x)$  in the look-up table directly from the input value  $x$  – it means assigning an arbitrary independent variable  $x$  to an integer interval  $i$ . First of all, consider a uniform grid of the input vector with the step denoted by  $\Delta$  with these properties: the step  $\Delta$  must be sufficiently small to capture the shape of the function. The very small  $\Delta$  can also be used for the approximation of discontinuous function which is transferred into a very steep continuous function. The variable  $x$  can be a possibly negative fractional number while the interval  $i$  must be a nonnegative integer. Therefore, we introduce the mapping function

$$i = \lfloor mx \rfloor + o \quad (4.9)$$

where  $m$  denotes the multiplier constant and  $o$  the interval offset and  $\lfloor \cdot \rfloor$  is floor function. Values of  $m$  and  $o$  will depend on interpolation break points in such way that if  $x$  is the break point, then the term  $mx$  will be an integer number. The multiplier constant can be found as  $m = \frac{1}{\Delta}$ . The vector  $\mathbf{x}_{\text{breaks}}$  which holds interpolation break point values and is used for the precomputation of the nonlinear function is constructed according to

$$\mathbf{x}_{\text{breaks}} = \mathbf{n}\Delta \quad (4.10)$$

where

$$\mathbf{n} = \{N_{\min}, N_{\min} + 1, \dots, -1, 0, 1, \dots, N_{\max} - 1, N_{\max}\} \quad (4.11)$$

where  $N_{\min} = \lfloor \frac{x_{\min}}{\Delta} \rfloor$ ,  $N_{\max} = \lfloor \frac{x_{\max}}{\Delta} \rfloor + 1$  and the offset  $o = -N_{\min}$ . When the vector  $\mathbf{x}_{\text{breaks}}$  is known, the precomputation of the nonlinear function for  $\mathbf{x}_{\text{breaks}}$  values proceeds. After the precomputation of the nonlinear function, interpolation coefficients can be computed, if the spline interpolation is chosen. The processing using the uniform grid is the following:

1. interval computation  $i = \lfloor mx \rfloor + o$ ,
2. fractional part computation  $x_p = x - x_{\text{breaks}}[i]$ ,
3. and finally the interpolation  $y = f(x_p)$ .

To compare computational costs of each interpolation, all the interpolation algorithms were implemented in C language according to pseudocodes in Appendix B with input parameters  $i$  and  $x$  and which returns interpolated function value.

The computational cost of all algorithms is compared in Figure 4.2 and Table 4.1. Figure 4.2 shows real conditions – time which was needed for the computation of one million of interpolations. It was tested on 2.6 GHz Intel processor. On the contrary, Table 4.1 shows the number of mathematical instructions per algorithm. Reader may notice different results. The cubic spline interpolation type 1 is in real conditions faster than the linear interpolation although they have the same number of mathematical operations. Further, the computational cost of the Newton polynomial is more than two times slower than the Hermite cubic interpolation in real conditions but there is not as big difference in the number of mathematical operations. These differences are caused by the proximity of division operations in algorithms and it confirms the fact that the mathematical division is not suitable operation for the efficient implementation of algorithm.

Algorithms can also be compared with regards to memory demands. The linear interpolation, the Newton polynomial interpolation and the Hermite interpolation are so called local interpolations. It means that they require only function values stored in look-up tables. The cubic spline interpolation of type 2 and 3 needs also precomputed second derivatives and memory demands are therefore two times higher. The cubic spline interpolation type 1 has the biggest memory demands. It requires storing of four coefficients per spline interval and it has therefore four times higher memory demands. If there is enough memory, this type of interpolation is the most convenient because it offers the most efficient implementation as well as  $C^2$  smoothness. If the memory demands are limited, one has to chose from other types of algorithms according to available memory and computational power.

Table 4.1: Computational cost comparison of interpolation algorithms.

Alg.	Spline 1	Spline 2	Spline 3	Newton	Hermite	Linear
Math. oper.	7	18	19	27	26	7

### 4.3.1 Non-uniform Grid Interpolation

The uniformly gridded interpolation has the advantage of the direct computation of the interpolation interval but it is not very convenient for the interpolation of sharp nonlinear transfer functions. It is because of a very small step of the grid that is required to capture the shape of the transfer function, which leads to using of big look-up tables. In contrast, the non-uniformly gridded interpolation contains more data points and smaller step of the grid in steep parts of the transfer function while sampling of almost linear parts of the transfer function can be quite sparse.

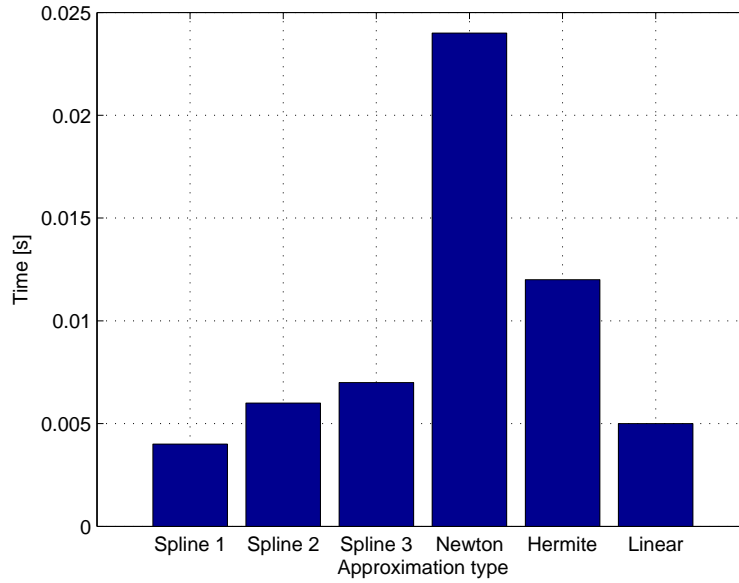


Figure 4.2: Computational cost of interpolation algorithms. Measured for one million of interpolations.

When using the non-uniformly gridded interpolation, some points from the original uniformly spaced breaks vector  $\mathbf{x}_{\text{breaks}}$  are removed, yielding more complicate interval computation. If the origin approach of the index computation introduced by (4.9) is used, the interval indexes are not consistent with the new ones anymore. Nevertheless, this problem can be solved by the introduction of a simple interval mapping function given by the vector  $f_{\text{mapping}}$  which will translate interval indexes from the uniform grid allowing direct access to data to the nonuniform grid in such way that several original indexes will belong to one new index. The new computational scheme for the 1-D interpolation is

1. interval computation  $j = \lfloor mx \rfloor + o$ ,
2. interval translation  $i = f_{\text{mapping}}[j]$ ,
3. fractional part computation  $x_p = x - x_{\text{breaks}}[i]$ ,
4. and finally the interpolation  $y = f(x_p)$ .

This modification enables the reduction of the look-up table while there is only one extra assignment  $i = f_{\text{mapping}}[j]$ . The drawback is that extra memory for mapping data is required, but mapping functions are one dimensional even for more dimensional spline interpolations, which will be exploited later.

The difficult task is the determination which data can be removed without losing the accuracy of the interpolation. Unimportant data can be found by the sequential removing of data points and testing of the impact to the interpolated transfer function.

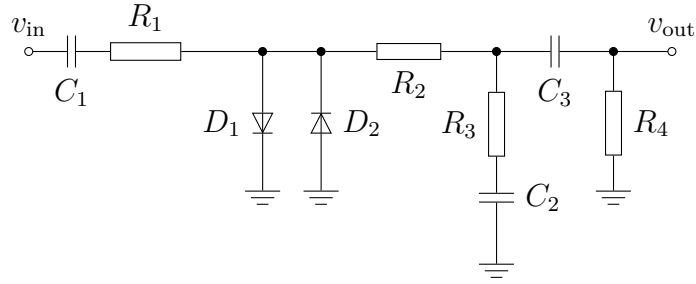


Figure 4.3: Circuit schematic of the nonlinear core of the Distortion effect.

The ideal combination of the reduced original data can be found if all combination of data points for all lengths  $2, 3, \dots, N$  where  $N$  is the length of original breaks vector are examined. The number of combinations for one reduced data length  $K$  is given by the binomical coefficient and the total number for all possible vector lengths given by  $2^{(N-2)} - 1$ . As it can be seen, this approach leads to the combination explosion with the increasing length  $N$  and therefore this approach is not computable for higher  $N$ .

In order to reduce the number of data points somehow, the algorithm given by the pseudo-code in Algorithm 1 is introduced. There is a constraint given by the constant access to data – the data must remain on positions of the regular grid and cannot be scattered. The algorithm removes unimportant data points  $x, f$  from the regular grid. It requires testing data  $x_{\text{test}}$  and interpolated function values  $f_{\text{test}}$  for given input  $x$  which are computed using the interpolation above the full set of data points. The algorithm goes through all data points except the boundary points. Each point is removed and the interpolation for all input data  $x$  is executed. Then the data point with the smallest error between the interpolation above the reduced set of data points and full data points is removed. After that, next iterations above the reduced data points are performed until the error is equal to the given maximal approximation error  $e_{\text{max}}$ . Finally, the mapping function between the uniform and nonuniform grid must be determined. The algorithm has quadratic computational complexity  $O(N^2)$ .

The algorithm was tested on the nonuniform grid piece-wise approximation of the transfer function of the guitar distortion effect Boss Metal zone. The circuit of the nonlinear distortion block, which is known as the diode clipper circuit, is given by the schematic in Figure 4.3 and circuit component values in Table 4.2. If the DK-method analysis is exploited, the nonlinear transfer function is one dimensional. All mentioned types of interpolation techniques were used to find which type of the interpolation needs the lowest amount of data. The nonlinear function was precomputed with values between  $-20$  and  $20$  with the step  $0.25$  and total number of data points is  $161$ . The chosen approximation error for Algorithm 1 was  $0.005$ .



```

input :  $x_{\text{test}}, x, f, e_{\text{max}}$ 
output :  $x_n, f_n$ 

 $f_{\text{test2}} \leftarrow \text{interpolate}(x_{\text{test}}, x, f)$ ;
 $x_n \leftarrow x$ ;
 $f_n \leftarrow f$ ;
 $N \leftarrow \text{length}(x_n)$ ;
while  $e < e_{\text{max}}$  do
  for  $i \leftarrow 1$  to  $N - 2$  do
     $x_r \leftarrow \text{removePointFromVector}(i, x_n)$ ;
     $f_r \leftarrow \text{removePointFromVector}(i, f_n)$ ;
     $f_{\text{test2}} \leftarrow \text{interpolate}(x_{\text{test}}, x_r, f_r)$ ;
     $e_{\text{vec}}[i] \leftarrow \max(f_{\text{test2}} - f_{\text{test2}})$ ;
  end
   $e, p \leftarrow \min(e_{\text{vec}})$ ;
   $x_n \leftarrow \text{removePointFromVector}(p, x_n)$ ;
   $f_n \leftarrow \text{removePointFromVector}(p, f_n)$ ;
   $N \leftarrow N - 1$ 
end

```

**Algorithm 1:** Nonuniform 1-D data reduction.

Table 4.2: Element values for schematic on Figure 4.3

$R_1$	$R_2$	$R_3$	$R_4$	$C_1$	$C_2$	$C_3$
2.2 k $\Omega$	10 k $\Omega$	4.7 k $\Omega$	100 k $\Omega$	10 $\mu$ F	15 nF	1 $\mu$ F

Results of the data reduction are available in Table 4.3 and Figures 4.4, 4.5, and 4.6. Table 4.3 shows the number of data points which are needed by each interpolation algorithm to give the error lower than 0.005. The lowest number of data points is needed by the cubic spline interpolation type 1 and 2, but these interpolations requires additional data to construct the spline (see the row: coefficients, the cubic spline type 1 requires four coefficients per interval and the cubic spline type 2 requires two coefficients per interval). The lowest number of data for the interpolation is needed by the Newton polynomial interpolation but this algorithm has the highest computational cost and it does not offer  $C^1$  smoothness.

Considering trade-off between the required data amount and the computational cost, the best approximation is provided by the cubic spline type 2. Quality of the simulation can be seen from Figures 4.4, 4.5. The first one shows the original precomputed function and the approximated transfer function for all interpolation

algorithms. The second figure shows the difference between the original transfer function and the approximated transfer function. All approximated functions have error smaller than the given maximal error 0.005. The average error is shown in Figure 4.6. It shows that the linear interpolation has the highest average error. Surprisingly, the Hermite interpolation provides the lowest average error although it offers only  $C^1$  continuity. It is because of the higher number of data points and also because of the higher error of the spline interpolation at boundaries of the transfer functions. This error could be probably lowered by use of different boundary conditions and thus different type of cubic spline interpolation. This simulation used the natural cubic spline with zero second derivatives at spline boundaries.

Table 4.3: Comparison of the number of data points required for approximation of diode clipper circuit.

Alg.	Spline 1	Spline 2	Spline 3	Newton	Hermite	Linear
Points	15	15	29	21	26	34
Coefficients	60	30	58	21	26	34

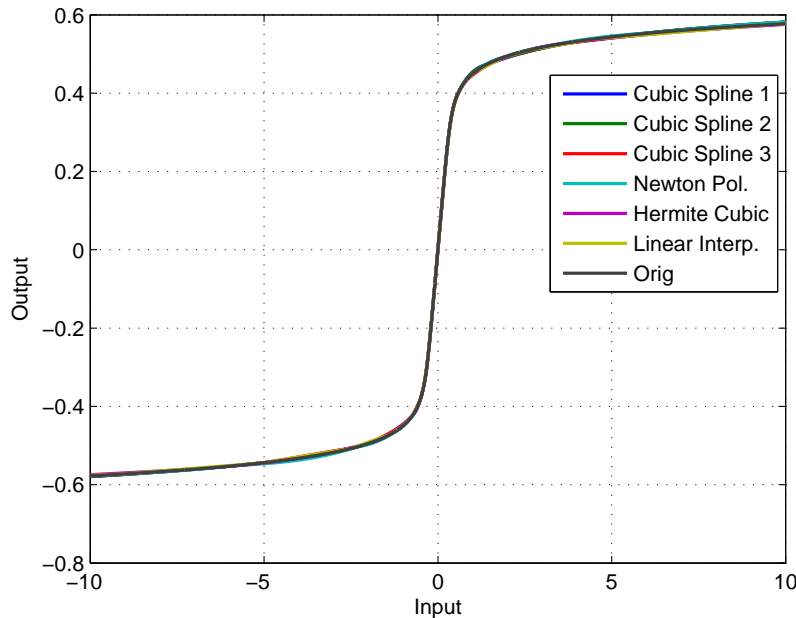


Figure 4.4: Approximation of the transfer function of the diode clipper circuit.

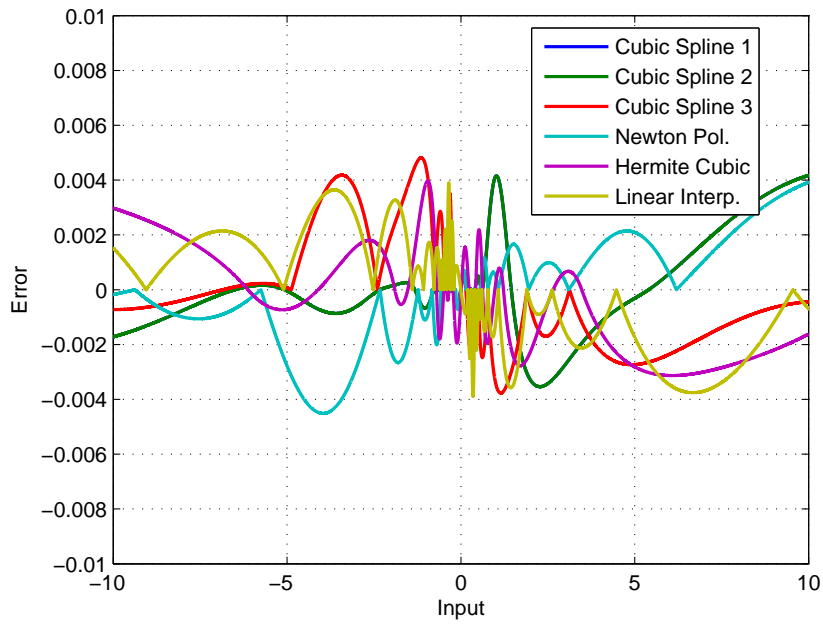


Figure 4.5: Error of the approximations of the transfer function of the diode clipper circuit.

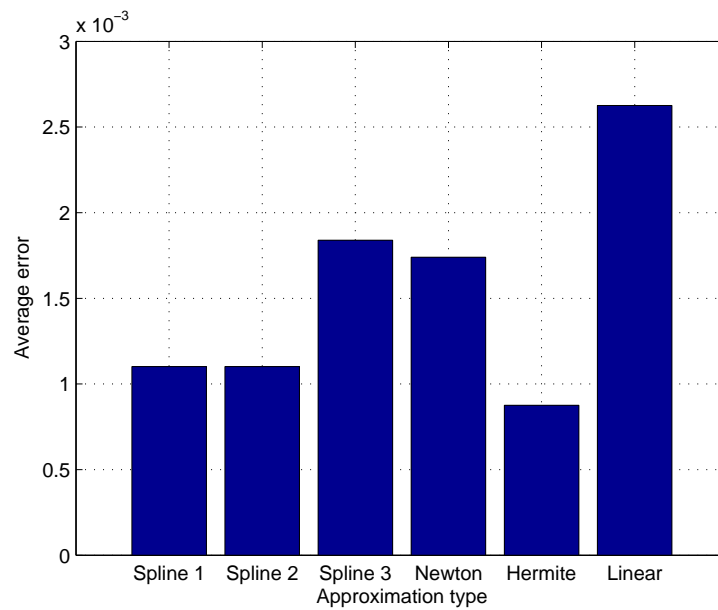


Figure 4.6: Average error of the approximations of the transfer function of the diode clipper circuit.

## 4.4 Approximation of N-D function

Algorithms for approximation of the 1-D nonlinear transfer function are not sufficient for most of simulated audio effect circuits and therefore, they must be extended into more dimensions. This extension can be made by the recursive calling of the 1-D interpolation function above multi-dimensional data and the dimension is being lowered until the last 1-D interpolation function is called. For instance, the 2-dimensional linear interpolation computes two 1-D linear interpolations

$$\begin{aligned} f(x_i) &= \text{linInterp}(f(x_i, y_i), f(x_i, y_{i+1})), \\ f(x_{i+1}) &= \text{linInterp}(f(x_{i+1}, y_i), f(x_{i+1}, y_{i+1})) \end{aligned}$$

for the variable  $y$  and then one 1-D linear interpolation is computed from the results

$$f(x, y) = \text{linInterp}(f(x_i), f(x_{i+1}))$$

for  $x$  variable in the next iteration. The  $N$ -dimensional linear interpolation requires recursive calling of

$$N_{\text{interp}} = 2^N - 1 \quad (4.12)$$

1-D linear interpolation in  $N$  iterations. The same principle can be exploited for other interpolation algorithms. The Hermite cubic and the Newton polynomial 1-D interpolation work with quartet of data points and therefore the 2-D interpolation must compute four 1-D interpolations in the first iteration. The total number of 1-D interpolations is

$$N_{\text{interp}} = \frac{4^N - 1}{3} \quad (4.13)$$

for the  $N$ -dimensional interpolation.

A little different situation is with spline interpolations because they work with more data (spline coefficients) than just function values. The spline interpolation in more dimensions is available using the tensor product. In case of 2-D splines, the interpolation is in the form of

$$f(x, y) = \sum_{i=0}^3 \sum_{j=0}^3 c_{i,j} x^i y^j \quad (4.14)$$

where  $c_{i,j}$  are spline coefficients. Totally, 16 spline coefficients are required for one function evaluation. The first iteration computes four 1-D spline interpolations with coefficients  $c_{i,j}$  and the variable  $y$  producing the new set of coefficients  $c_i$ . The second iteration computes one 1-D spline interpolation with coefficients  $c_i$  and the variable  $x$ . Generally,  $\frac{4^N-1}{3}$  1-D spline interpolations must be computed for the  $N$ -dimensional cubic spline interpolation and

$$N_{\text{coef}} = 4^N \quad (4.15)$$

spline coefficients must be stored in memory for one spline interval. The total number of coefficients is given by

$$N_{\text{coef}} = 4^N \prod_i^N N_i \quad (4.16)$$

where  $N_i$  is the number of intervals in the break points vector in  $i$ -th dimension.

Similarly to the 1-D cubic spline interpolation, not all spline coefficients must be stored. The 1-D spline interpolation must know function values and continuous second derivative values to compute all spline coefficients. Because the function is multivariate, the continuous second derivative consists of partial derivatives  $c_x(x, y) = \frac{\partial f(x, y)}{\partial x^2}$ ,  $c_y(x, y) = \frac{\partial f(x, y)}{\partial y^2}$  and cross derivative  $c_{xy}(x, y) = \frac{\partial f(x, y)}{\partial x^2 \partial y^2}$  in case of the 2-D function and similarly in higher dimensions. If we denote these derivatives and the function values as coefficients, the number of coefficients which must be stored is

$$N_{\text{coef}} = 2^N \prod_i^N N_i \quad (4.17)$$

which is  $2^N$  times lower than if full spline coefficients are stored. The computational scheme for the 2-D interpolation is following. Firstly, spline coefficients for the variable  $y$  are computed from stored function values  $f(x, y)$  and derivative  $c_y(x, y)$  and the evaluation of the spline gives new function values  $f(x)$  to be interpolated for the variable  $x$ . Similarly, the derivative  $c(x)$  is evaluated from spline coefficients computed from the stored partial derivative  $c_x(x, y)$  and the cross derivative  $c_{x,y}(x, y)$ . The first iteration computes four unknown spline coefficients and two spline interpolations. The second iteration computes two unknown spline coefficients from values  $f(x)$  and  $c(x)$  computed in the previous iteration and one spline interpolation to get the final interpolated function value. The total number of 1-D spline interpolations needed by the  $N$ -dimensional interpolation is  $2^N - 1$ .

Multi-dimensional versions of algorithms were compared with regards to memory demands and the computational cost. Figure 4.7 shows the dependency on the number of mathematical operations on interpolation dimension for each algorithm. The lowest number, except the 1-D interpolation, has the linear interpolation. The cubic spline interpolation of type 1 is useful up to the 2-D interpolation, other types of spline interpolation offer lower computational costs for higher number of dimension. This due to fact that, although they have to compute spline coefficients during the interpolation, they compute only  $2^N - 1$  of 1-D spline interpolations while the cubic spline 1 computes  $\frac{4^N - 1}{3}$  1-D spline interpolations.

Figure 4.8 shows memory demands for the  $N$ -th dimension of each algorithm. Local interpolation algorithms – the linear interpolation, the cubic Hermite and the Newton polynomial interpolations – have the lowest memory demands. The number of points in break points vectors in each dimension is 20. Considering the size of

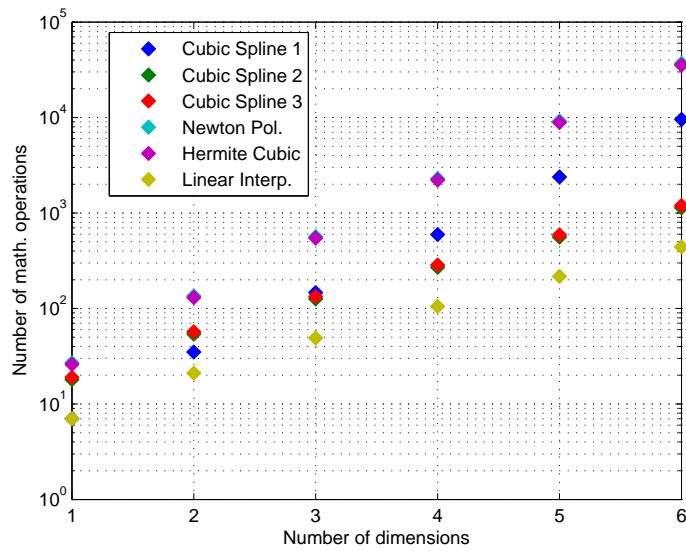


Figure 4.7: Number of operations required by  $N$ -dimensional interpolation.

look-up tables on the order of megabytes, these interpolations are capable of the 5-D interpolation if break vectors have around 20 break points. Spline interpolations can be used maximally for 3-D or 4-D interpolations, depending on the number of break points.

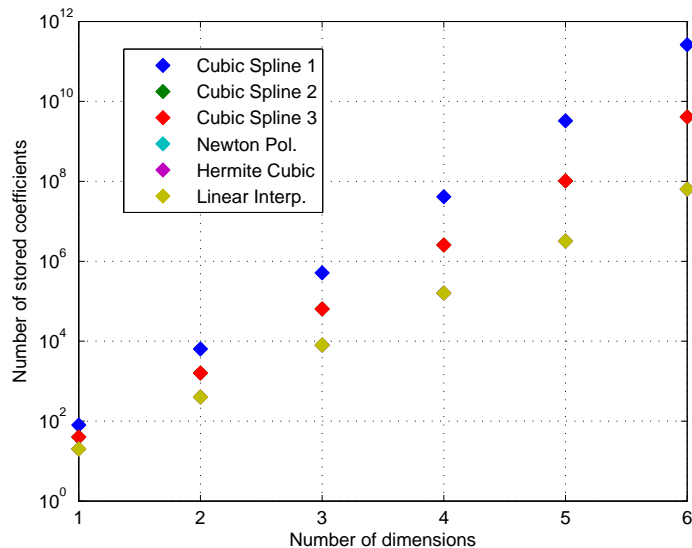


Figure 4.8: Number of coefficients to be stored for  $N$ -dimensional interpolation. Spline interpolation 2 and 3 are the same as well as Linear, Hermite and Newton are the same.

Real conditions were also tested. Figure 4.9 shows the computational cost for one million interpolations and numbers of interpolations, which can be computed in real-time for one sample with sampling frequencies 48 kHz and 192 kHz on 2.6 GHz Intel processor, are given in Table 4.4 and 4.5.

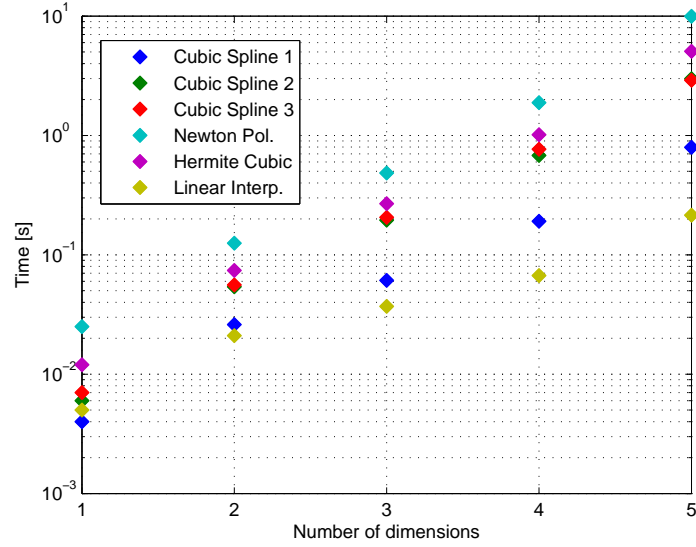


Figure 4.9: Computation time of one million  $N$ -D interpolations.

Table 4.4: Maximal number of interpolations per sample for  $f_s = 48$  kHz.

Dim.	Spline 1	Spline 2	Spline 3	Newton	Hermite	Linear
1-D	5210	3470	2980	833	1740	4170
2-D	801	386	372	167	282	992
3-D	342	107	101	43	77	563
4-D	109	30	27	11	20	311
5-D	26	6	7	2	4	96

Results show that the lowest computational cost is provided by the linear interpolation and the cubic spline interpolation type 1. The dependence of the spline interpolation real computational cost on the interpolation dimension is in contrast with the theoretical computational cost given in Figure 4.7. The real computational cost does not increase as rapidly as the theoretical and the cubic spline interpolation type 1 provides the lower computational cost than other spline interpolations. This is because of the compiler parallel optimization (see chapter 4.4.2 for details). It

Table 4.5: Maximal number of interpolations per sample for  $f_s = 192$  kHz.

Dim.	Spline 1	Spline 2	Spline 3	Newton	Hermite	Linear
1-D	1300	868	744	208	434	1040
2-D	200	96	93	41	70	248
3-D	85	26	25	10	19	141
4-D	27	7	6	2	5	77
5-D	6	1	1	0	1	24

is also obvious that the maximal dimension to work in real time is three or four, especially if higher sampling frequencies are used, which is desired to suppress the aliasing distortion.

#### 4.4.1 Non-uniform grid interpolation

The previous chapter showed that the number of interpolation coefficients to be stored increases rapidly with increasing dimension of interpolation algorithm. Therefore, the number of break points in break points vectors must be as low as possible. The higher order interpolation can make use of the non-uniform interpolation with constant access to data in the same way as the 1-D interpolation. The break points must lay on the original uniformly spaced grid. Some of them can be removed but they still have to build the regular grid although non-uniformly spaced. Further, mapping functions for breaks vectors are required for each dimension. These mapping vectors are however only one dimensional. The computational scheme is similar to the 1-D interpolation. For instance, the 2-D spline interpolation type 1 is computed according to:

1.  $i = \lfloor m_x x \rfloor + o_x$ ,
2.  $j = \lfloor m_y y \rfloor + o_y$ ,
3.  $x_{\text{part}} = x - x_{\text{breaks}}[i]$ ,
4.  $y_{\text{part}} = y - y_{\text{breaks}}[j]$ ,
5.  $a = ((c_{1,i,j}y_p + c_{2,i,j})y_p + c_{3,i,j})y_p + c_{4,i,j}$ ,
6.  $b = ((c_{5,i,j}y_p + c_{6,i,j})y_p + c_{7,i,j})y_p + c_{8,i,j}$ ,
7.  $c = ((c_{9,i,j}y_p + c_{10,i,j})y_p + c_{11,i,j})y_p + c_{12,i,j}$ ,
8.  $d = ((c_{13,i,j}y_p + c_{14,i,j})y_p + c_{15,i,j})y_p + c_{16,i,j}$ ,
9.  $f = ((ax_p + b)x_p + c)x_p + d$ .

A difficult task again is to find which data can be removed without losing the accuracy of the transfer function approximation. It is more complicated than the 1-D case because while some data points are being removed from the input vector in case



of the 1-D interpolation, whole rows or columns must be removed from the function values matrix in case of the 2-D interpolation and similarly in higher dimensions, let us say that a  $N - 1$  dimensional subarray is being removed from the given  $N$ -D array of function values. Otherwise the algorithm for data reduction is similar to its 1-D version and is given by pseudocode in Algorithm 2.

```

input :  $x_{\text{test}}, x, f, e_{\text{max}}, dim$ 
output :  $x_n, f_n$ 
 $f_{\text{test2}} \leftarrow \text{interpolate}(x_{\text{test}}, x, f);$ 
 $x_n \leftarrow x, f_n \leftarrow f;$ 
while  $e < e_{\text{max}}$  do
  for  $idim \leftarrow 1$  to  $dim$  do
    for  $i \leftarrow 1$  to  $\text{length}(x_n[idim]) - 2$  do
       $x_r[idim] \leftarrow \text{removePointFromVector}(i, x_n[idim]);$ 
       $f_r \leftarrow \text{removeSubArray}(i, f_n, idim);$ 
       $f_{\text{test2}} \leftarrow \text{interpolate}(x_{\text{test}}, x_r, f_r);$ 
       $e_{\text{vec}}[i] \leftarrow \max(f_{\text{test2}} - f_{\text{test2}});$ 
    end
  end
   $e, p, idim, \leftarrow \min(e_{\text{vec}});$ 
   $x_n[idim] \leftarrow \text{removePointFromVector}(p, x_n[idim]);$ 
   $f_n \leftarrow \text{removeSubArray}(p, f_n, idim);$ 
end

```

**Algorithm 2:** Nonuniform N-D data reduction.

#### 4.4.2 Parallel evaluation of interpolations

The computation of the  $N$ -D interpolation consists of the computation of multiple 1-D interpolations. For instance, considering the 2-D cubic spline interpolation with the computational scheme given in the previous section, it consists of four independent 1-D interpolations with different spline coefficients but the same variable  $y$ . This task is very convenient for the parallel computing using the Single Instruction Multiple Data (SIMD) processing, which means the same mathematical operation with multiple data. As a result of the parallelization, these four 1-D interpolations can be computed at the same time reducing the computational cost of the interpolation. The higher dimension of the interpolation, the higher number of 1-D interpolation can be parallelized and it depends only on the DSP system architecture how many of parallel computation units are available. On the other hand, it means that the interpolation algorithm code must be optimized for the given DSP system architecture and the

given processor and it can not be universal if the best performance has to be achieved. For instance, we will consider ordinary modern personal computer processors. The SIMD technology is available via the Streaming SIMD Extensions (SSE) instruction set extension [94]. This extension supports the horizontal as well as the vertical parallel computation of operands stored in processor registers. The SIMD registers are 128 bit. That means, when working with floating point numbers, four coefficients in single precision or two coefficients in double precision floating point can be stored in the register as one packed data word. If the vertical SSE instruction is called with two operands (coefficients in packed data words) stored in SIMD registers, then two or four instruction are executed simultaneously from stored coefficients point of view. Let us consider the 2-D cubic spline interpolation again. Spline coefficients are stored in the memory. There are 16 coefficients per spline interval determined by indexes  $i, j$ . The first iteration of the cubic spline computes following four 1-D spline interpolations

$$\begin{aligned} a &= ((c_{1, i,j}y_p + c_{2, i,j})y_p + c_{3, i,j})y_p + c_{4, i,j}, \\ b &= ((c_{5, i,j}y_p + c_{6, i,j})y_p + c_{7, i,j})y_p + c_{8, i,j}, \\ c &= ((c_{9, i,j}y_p + c_{10,i,j})y_p + c_{11,i,j})y_p + c_{12,i,j}, \\ d &= ((c_{13,i,j}y_p + c_{14,i,j})y_p + c_{15,i,j})y_p + c_{16,i,j}. \end{aligned}$$

The mathematical operation which are aligned vertically can be computed simultaneously using three add operations and two multiply operations. This however requires proper alignment of coefficients in the memory. The alignment of coefficients for one spline interval is following:

$$\left[ c_1 \ c_5 \ c_9 \ c_{13} \ c_2 \ c_6 \ c_{10} \ c_{14} \ c_3 \ c_7 \ c_{11} \ c_{15} \ c_4 \ c_8 \ c_{14} \ c_{16} \right]$$

As a result, the whole 2-D cubic spline interpolation consists of two 1-D interpolations instead of original five. This implementation is possible also for higher dimension, only the coefficients must be properly aligned in memory. The 3-D interpolation consists of six 1-D interpolations instead of original 21, which means that theoretically almost 4 times lower computational cost can be achieved but the real implementation can contain little bit higher computational overhead connected with moving of operands into SIMD registers.

A similar parallelization can also be made with other types of interpolations. Each algorithm however can require different memory organization. But generally, coefficients are stored according to the following scheme (3-D interpolation):

$$\left[ \dots \ c_{i,j,k} \ c_{i+1,j,k} \ \dots \ c_{i+N_1,j,k} \ c_{i,j+1,k} \ \dots \ c_{i+N_1,j+N_2,k} \ \dots \ c_{i,j,k+1} \ \dots \right]$$

It means that the coefficients along the first dimension are neighbouring although the interpolation scheme is opposite – from the last dimension to the first one.

## 4.5 Customized approximation of transfer function

The approximation of the transfer function is not restricted to use only one approximation technique but one can use also a combination of different algorithms. For instance, the shape of the transfer function can be dependent on additional parameters like the power supply voltage value. If this dependency is linear (the saturation voltage equal to the power supply voltage value), the resulting transfer function can be computed as a linear combination of two transfer functions defined for boundary values of the power supply voltage while the relation between the input and the output is given e.g. by the cubic spline interpolation. Then, for given two inputs - the input signal value and e.g. the fluctuating power supply voltage (the power sagging effect known from guitar tube power amplifiers [18]), two spline interpolations and one linear interpolation are computed. It is also possible to compute the linear interpolations between cubic spline coefficients and then compute one cubic spline interpolation.

This approach can also be used for approximation of the parametric circuits transfer function. While there is requirement to have a continuous approximation with regards to input variables, it is sufficient to use the linear interpolation between approximated transfer functions for different circuit parameter values. This enables reduction of the amount of the interpolation data. The linear interpolations between the transfer function coefficients do not have to be computed during the evaluation of function value but they can be computed only when the parameter is changed, which allows further reduction of the computational cost.

### 4.5.1 Reshaping of transfer function

Parametric circuits often have the transfer function with very similar shape for different circuit parameter settings and the transfer function often seems to be scaled at the first sight. But not just only parametric circuits. Figure 4.10 shows the transfer function of the triode tube preamp from Figure 4.1. One can see that the transfer function is only scaled and shifted with the capacitor state voltage  $v_{c1}$ .

To describe the shifted and scaled transfer function, the reshaping function is introduced in form

$$\hat{f}_2(x) = m_1 + m_2 f_1(m_3 + m_4 x) \quad (4.18)$$

where  $f_1(x)$  is the stored transfer function,  $x$  is the independent input variable and  $m_1$ ,  $m_2$ ,  $m_3$  and  $m_4$  are scaling coefficients which allow the reshaping of the stored transfer function  $f_1(x)$  and which are dependent on additional inputs -  $v_{c1}$  in case of the transfer function from Figure 4.10. The reshaping coefficients can be found by

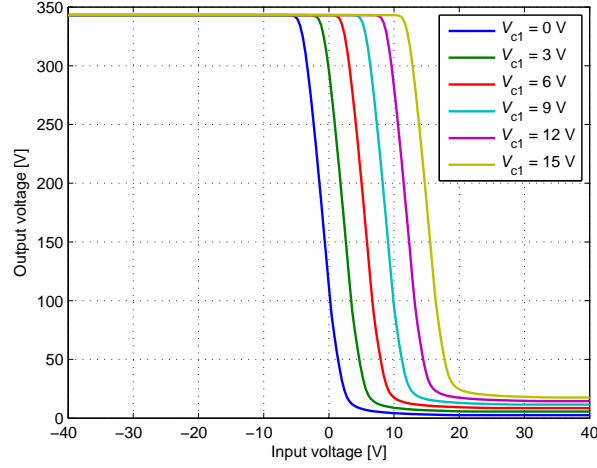


Figure 4.10: The transfer function of the tube preamp

finding of the least square error between the rescaled function  $\hat{f}_2(x)$  from the stored function  $f_1(x)$  and the target function  $f_2(x)$  using the error function

$$\begin{aligned}
 f_{\text{err}} &= \left( \sum_{i=0}^{N-1} f_2(x_i) - \hat{f}_2(x_i) \right)^2 \\
 &= \left( \sum_{i=0}^{N-1} f_2(x_i) - (m_1 + m_2 f_1(m_3 + m_4 x_i)) \right)^2.
 \end{aligned} \tag{4.19}$$

Now the task is to find coefficients  $m_1$ ,  $m_2$ ,  $m_3$  and  $m_4$  in such way that they will minimize the error function (4.19). We can suppose that there is only one optimal solution of  $\min(f_{\text{err}})$  and this solution lies at the stationary point of (4.19). The stationary point is found by solving

$$\mathbf{0} = \nabla f_{\text{err}}(m_1, m_2, m_3, m_4) \tag{4.20}$$

where the partial derivatives are

$$\frac{\partial f_{\text{err}}}{\partial m_1} = 2 \sum_{i=0}^{N-1} (-f_2(x_i) + m_1 + m_2 f_1(m_3 + m_4 x_i)), \tag{4.21}$$

$$\frac{\partial f_{\text{err}}}{\partial m_2} = -2 \sum_{i=0}^{N-1} ((f_2(x_i) - m_1 - m_2 f_1(m_3 + m_4 x_i)) f_1(m_3 + m_4 x_i)), \tag{4.22}$$

$$\frac{\partial f_{\text{err}}}{\partial m_3} = -2 \sum_{i=0}^{N-1} ((f_2(x_i) - m_1 - m_2 f_1(m_3 + m_4 x_i)) m_2 f_1'(m_3 + m_4 x_i)), \tag{4.23}$$

$$\frac{\partial f_{\text{err}}}{\partial m_4} = -2 \sum_{i=0}^{N-1} (((f_2(x_i) - m_1 - m_2 f_1(m_3 + m_4 x_i))) m_2 f_1'(m_3 + m_4 x_i) x_i). \tag{4.24}$$

Because the function  $f_1(x)$  is given by the look-up table, the interpolation of function values and numerical derivatives must be used to solve (4.20). The system of equations (4.20) is solved numerically using the Newton-Raphson method with residual functions given by (4.21, 4.22, 4.23, 4.24). If the shape of both functions  $f_1(x)$  and  $f_2(x)$  is similar, the good starting point for the Newton-Raphson method is  $\mathbf{m}_0 = \{0, 1, 0, 1\}$ . After obtaining coefficients  $m_1, m_2, m_3$  and  $m_4$ , the output function value can be evaluated from computational scheme:

1.  $x_2 = m_3 + m_4x$
2.  $i = \text{index}(x_2)$
3.  $y = (((a_ix_2 + b_i)x_2 + c_i)x_2 + d_i)m_4 + m_3$

if the cubic spline interpolation is used for the approximation of the transfer function  $f_1(x)$ .

Coefficients  $m_1, m_2, m_3$  and  $m_4$  are not constant but they depend on parameter values –  $v_{c1}$  in case of the transfer function from Figure 4.10. Therefore, the equation (4.20) is solved for different values of  $v_{c1}$  voltage and as a result, the dependency of coefficients  $m_1, m_2, m_3, m_4$  on  $v_{c1}$  voltage is shown in Figure 4.11 and the reshaped transfer function for  $v_{c1} = 15V$  from the transfer function for  $v_{c1} = 0V$  is displayed in Figure 4.12. One can observe almost linear dependency of coefficients  $m_1, m_2, m_3, m_4$  on  $v_{c1}$  voltage which can be expressed using equations

$$\begin{aligned}
 m_1 &= 1.0075v_{c1} - 0.0066 \\
 m_2 &= -0.0029v_{c1} + 1.0000 \\
 m_3 &= -1.1111v_{c1} + 0.1124 \\
 m_4 &= 0.0028v_{c1} + 0.9998.
 \end{aligned}
 \tag{4.25}$$

If the low value coefficients from (4.25) are omitted, the whole approximation of the tube preamp transfer function can be expressed as

$$\hat{f}_2(x) = 1.0075v_{c1} + f_1(-1.1111v_{c1} + x).
 \tag{4.26}$$

This reshaping technique allowed reduction of one interpolation dimension. Instead of originally required 2-D spline interpolation, 1-D interpolation can be used still with low error of the transfer function as it can be seen from Figure (4.12).

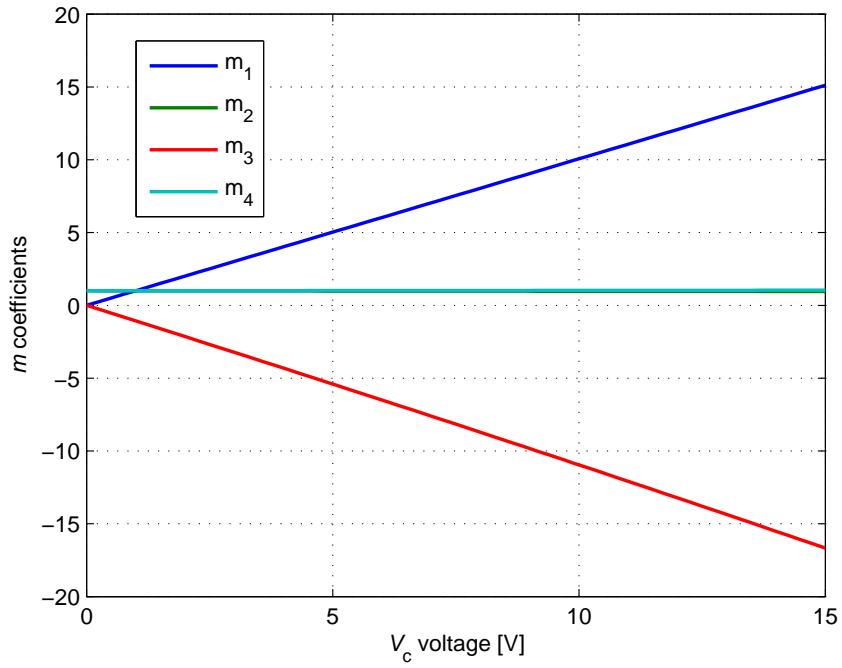


Figure 4.11: Reshaping coefficients dependency on cathode capacitor voltage

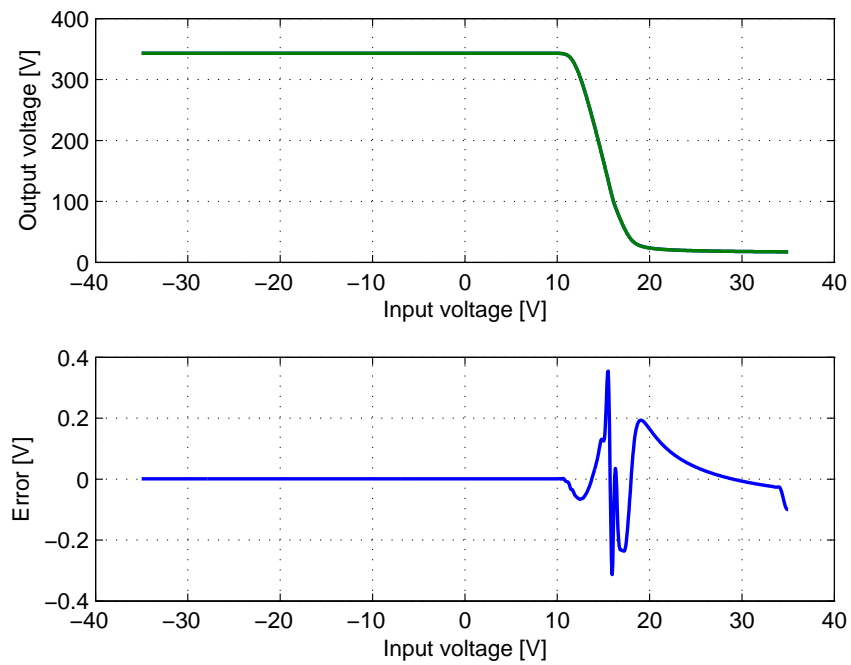


Figure 4.12: Reshaped transfer function of the tube preamp.

## 4.6 Summary

This chapter was focused on the use of approximations for the nonlinear function evaluation. The precomputation of the nonlinear circuit for several input variables is always the first step of the approximation of nonlinear circuit functions. Two types of approximations were examined. The first one is based on the direct precomputation and latter approximation of nonlinear ODEs. The whole circuit is in this case considered as the system with several inputs built by real inputs and states of energy-storing components. The advantage of this approach is that the approximating functions can be designed in such way that the real-time simulation algorithm is independent on the sampling frequency value and the computational cost is very low. The disadvantage of this approach is the fact that the dimension of the approximating function depends on the number of energy-storing components and real circuit inputs, which can be high even for quite simple audio circuits. This approach is therefore suitable for real-time simulation of circuits with two or three energy-storing components, which is e.g. the often simulated triode amplifier.

The second approach exploits the DK-method analysis of the given circuit. The approximating function dimension depends on the number of nonlinear current models in the circuit. If this number is higher than the number of energy-storing components, then this approach is inefficient. However, the analysis of DK-method inputs enabling reduction of the dimension was introduced.

The comparison of several approximation techniques in real-time conditions was the main focus and the biggest contribution of this chapter. The concrete use of the approximation algorithm however depends on the architecture of the DSP system on which the algorithm will be implemented. The spline interpolation and the linear interpolation are very efficient algorithms but the cubic spline interpolations require additional data (spline coefficients) to be stored. If memory demands are critical, local types of approximations (the linear or the cubic Hermite piece-wise interpolation) should be used. But especially the second one has quite high computational cost.

Memory demands can be reduced by use of non-uniform grid interpolation with constant access to data, which can, without any serious increasing of computational cost, significantly reduce the amount of data stored in the memory.

Generally, it can be stated that approximations up to four dimensions can be used for real-time simulations, but this number of course depends on the given DSP system. The use of approximations is not limited only for the evaluation of nonlinear circuit functions. It can also be used for simulation of linear parametric systems, e.g. the simulation of the guitar loudspeaker cabinet measured using microphone placed in a grid of measuring points in front of the loudspeaker [7].

## 5 COMPLEX SYSTEM SIMULATION

Circuits of analog audio effects are not usually as simple as in examples given in previous chapters. The typical guitar preamp or the power amplifier can consist up to ten nonlinear equations, which is not feasible to compute in real-time numerically and the approximation of the  $N$ -dimensional nonlinearity would require huge data sets of coefficients. The reasonable number of approximation dimensions is 4 as it was shown in the previous chapter and this number is not sufficient for the simulation of more complex systems.

Therefore, this chapter will deal with techniques which can be used for real-time simulations of more complex systems. They will differ in computational costs as well as in the accuracy of the simulation and thus they are suitable for implementation into different DSP systems. The base of all these techniques is a decomposition of the complex system into separate and simple blocks which are solved separately and then they are connected according to the signal flow of the simulated system. This technique has been commonly used in various papers [32, 22, 15]. The big advantage is the low computational cost and the decomposition into the blocks can be easily done for some systems as well. Some problems occur when there is a tight coupling between neighbouring parts of the circuit and one part strongly affects the other one and vice-versa. In these cases, this simple decomposition produces high error into the overall simulation.

Therefore, the main aim of discussed techniques is to preserve mutual interactions between separate blocks. These techniques should allow such decomposition that will provide the same results as without the decomposition or at least such decomposition which would be inaudible in the output signal.

This chapter will address three methods of the decomposition including case studies on different audio effects circuits.

### 5.1 Modified Block-Wise Method

This method was designed as the efficient and simple method enabling the simulation of the mutual interaction between adjoined blocks. The research of this method has been published in author's publications [8, 9, 10, 11]. Audio effect circuits are often built by a cascade of amplifiers, filters, nonlinear waveshapers, etc. There is therefore often coupling only between directly adjoined blocks and there is no direct coupling between e.g. the first and third block. We can suppose that there is a very small impact of the third block behavior to the first block. Hence, the simulation of the first block has to take account of the current, typically nonlinear, flowing into the second block and one has to simulate both first two blocks together. As



a result of this, the output impedance of the first block is matched with the input resistance of the adjoined second block and the output voltage or the output signal of the first block is correct. Up to this point, only the simulation of the first block was considered although the second block has been used and there is also available the output signal of the second block. However, this output signal is not matched with the third block and therefore, it cannot be used. The simulation of the second block requires the simulation of the second block together with the third one. The output signal between the first and second block is then used as the input signal for the connected second and third block. If there are more blocks connected in series, we can continue with the decomposition in similar way. The whole process of such decomposition is illustrated in Figure 5.1.

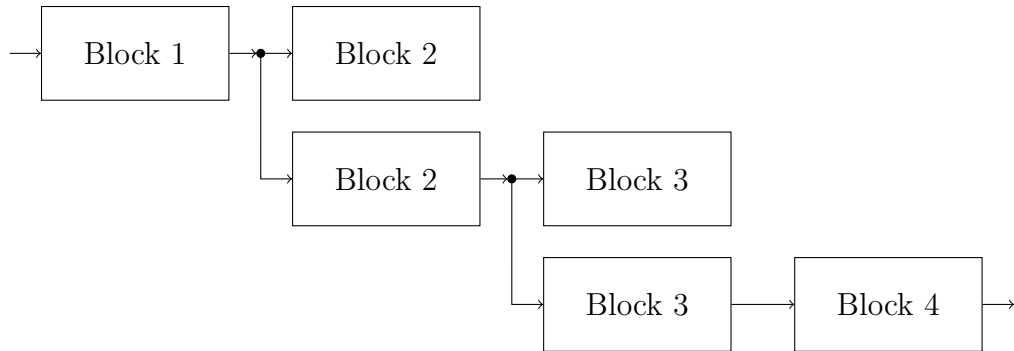


Figure 5.1: Example of decomposition into separate blocks using the modified block-wise method.

This approach is especially suitable for guitar preamp simulations. The guitar preamps usually contain up to five tube amplifier stages connected in series [87]. The circuit schematic of one typical tube amplifier stage with one triode is shown in Figure 5.2 with typical values in Table 5.1. Individual tube amplifier stages are characterized with the nonlinear input resistance which is able to affect the previous tube amplifier stage. The nonlinear input resistance of the typical tube amplifier stage is shown in Figure 5.3. The input resistance is foremost given by resistors  $R_{in}$ ,  $R_g$  and the grid current  $I_g$  of the tube. There is no grid current for negative grid-to-cathode voltages and there is only the current flowing through the series combination of resistors  $R_{in}$ ,  $R_g$  and hence, the input resistance is equal to the sum of resistances  $R_{in}$  and  $R_g$ . When the grid-to-cathode voltage becomes positive, the grid current  $I_g$  starts to flow into the tube and the tube input resistance starts to decrease very rapidly. When the tube input resistance gets lower with the increasing grid-to-cathode voltage, the parallel combination of the grid resistor  $R_g$  and the input tube resistance gets lower as well and the overall input resistance is built mainly only by the resistance  $R_{in}$ .

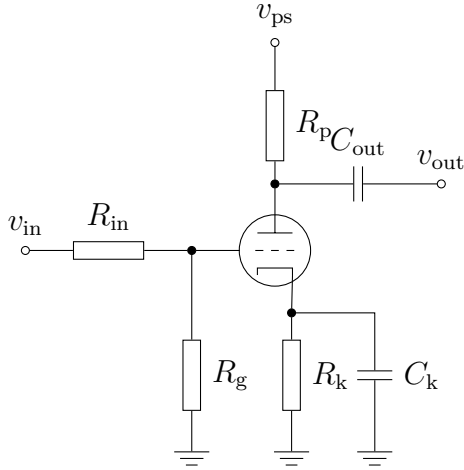


Figure 5.2: Circuit schematic for typical tube amplifier stage.

Table 5.1: Values for circuit components for typical tube amplifier stage.

$R_{in}$	$R_g$	$R_k$	$R_p$	$C_{out}$	$C_k$	$V_{ps}$
470 k $\Omega$	1 M $\Omega$	1.8 k $\Omega$	100 k $\Omega$	22 nF	1 $\mu$ F	350 V

Consequently, when this nonlinear resistance is connected to the plate circuit of the previous tube, it builds the nonlinear load for the previous tube. The effect of the nonlinear load to the tube amplifier stage can be examined e.g. by the simulation of this stage for different load resistor values, which is shown in Figure 5.4 where the load resistor has value of 0.5 M $\Omega$ , 1.0 M $\Omega$  and 4.0 M $\Omega$ . One can see the compression of the transfer function for smaller load resistance values and therefore, one can expect the compression of the positive half-wave of the tube amplifier stage output signal as it can be seen in Figure 5.5.

This figure shows the comparison of output signals for the tube amplifier stage with the constant load and the tube amplifier stage with the connected another tube stage which serves as the nonlinear load. The input signal was a sine wave with a frequency of 1 kHz and an amplitude of 10 V. As it was expected, the compression of the output signal occurred because of the grid current of the adjacent tube flowing through the plate resistor of the first tube. There is also the bias shift caused by charging and discharging of the decoupling capacitor connected between both tube stages by the grid current. This affects the operating point of the adjacent tube stage. These results show that the simple decomposition into separate tube stages with the constant load is inaccurate and thus the modified decomposition have to be used.

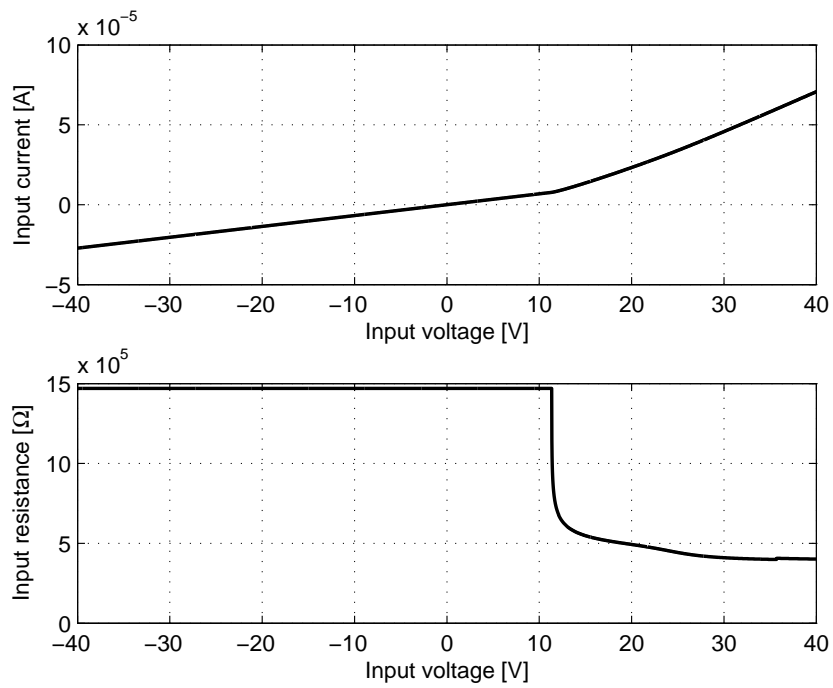


Figure 5.3: Input volt-ampere characteristics (top) and resistance (bottom) of tube amplifier stage.

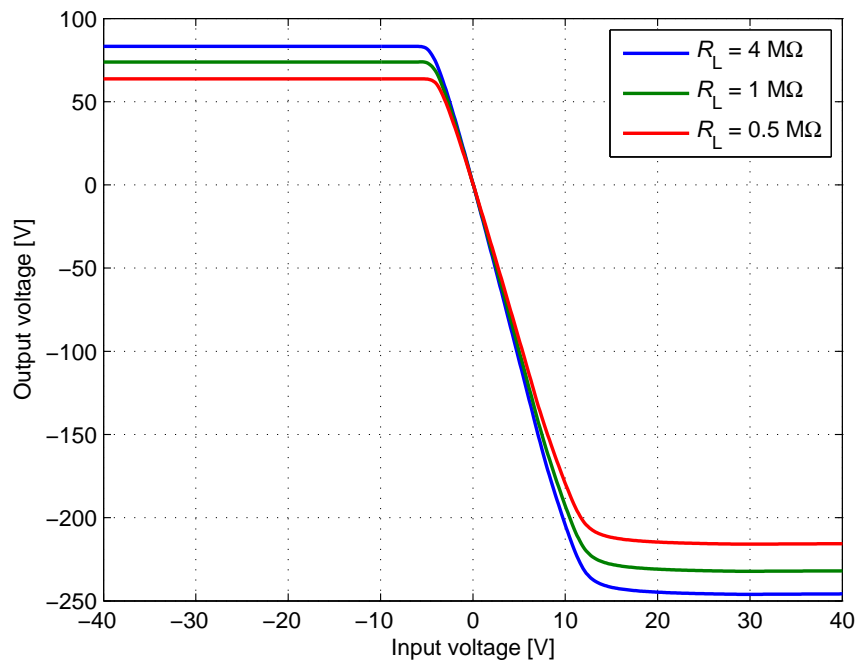


Figure 5.4: Transfer function of the tube amplifier stage with different load resistor.

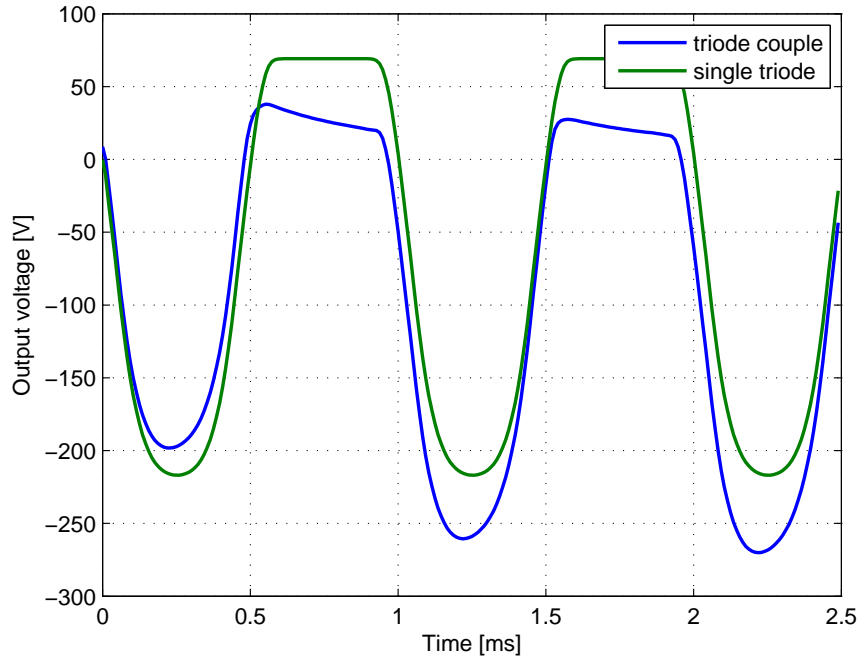


Figure 5.5: Transient analysis of the tube amplifier stages with the nonlinear load and the constant load.

The decomposition using the modified block-wise method supposes that there is no or very small interaction between the first and the third block and therefore, it is sufficient to consider only the second block terminated with the constant load as the nonlinear load for the first block. Because it is only a hypothesis, it should be proved by some simulations. Firstly, we need to know the impact of the second tube stage variable load on the output signal of the first tube stage. Two input stages given by the circuit schematic in Figure 5.2 were connected in series and the load resistor was connected to the output of the second stage. After obtaining the circuit equations, the simulation for the load resistor  $R_L = 5\text{ k}\Omega$ ,  $R_L = 50\text{ k}\Omega$ ,  $R_L = 500\text{ k}\Omega$  and  $R_L = 5\text{ M}\Omega$  was performed. Results are shown in Figure 5.6. The input signal was a sine wave with a frequency of 1 kHz and an amplitude of 10 V. The upper graph shows output signals of the first tube stage for different load resistor of the second stage and the lower graph shows the difference between them. The output signal obtained from the simulation with  $R_L = 5\text{ M}\Omega$  was chosen as the reference signal. As it can be seen from the graphs, there is an interaction between the second tube stage variable load and the first tube stage output signal. This interaction however manifests for rather small values of the load of the second tube stage, which is not the case of the connected third block with minimal input resistance given by the resistor  $R_{in}$ , as it can be seen from Figure 5.3. This value is usually high (hundreds

of  $k\Omega$  and in this case  $470\text{ k}\Omega$ ) and therefore, the nonlinear input resistance of the third tube stage has the minor impact to the output signal of the first tube stage. This was also proved in Figure 5.7 which shows the comparison of the plate output signals of the tube stage with one or two tube stages connected in series as the load of the first stage. The input signal was again the sine wave with the frequency of  $1\text{ kHz}$  and the amplitude of  $10\text{ V}$ . The upper graph shows output signals of the first tube stage and the lower part their difference. This simulation proved that this type of decomposition into blocks is very convenient for use with simulation of the guitar preamp.

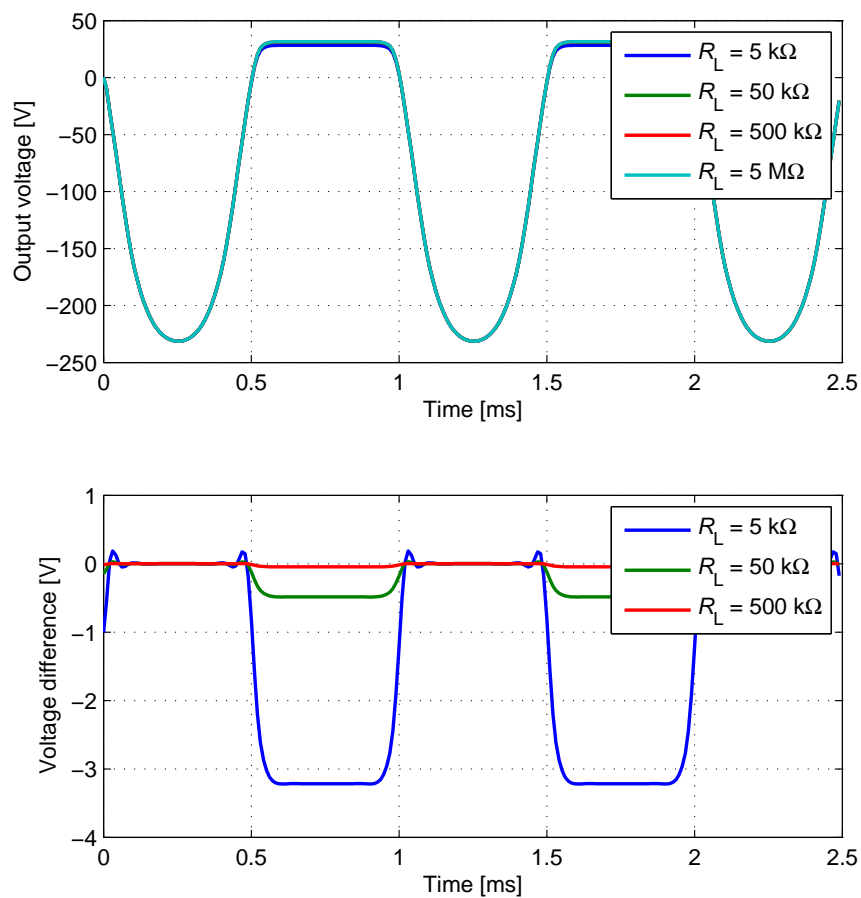


Figure 5.6: Comparison of the output signals for different load resistor in the tube stage connected as the nonlinear load.

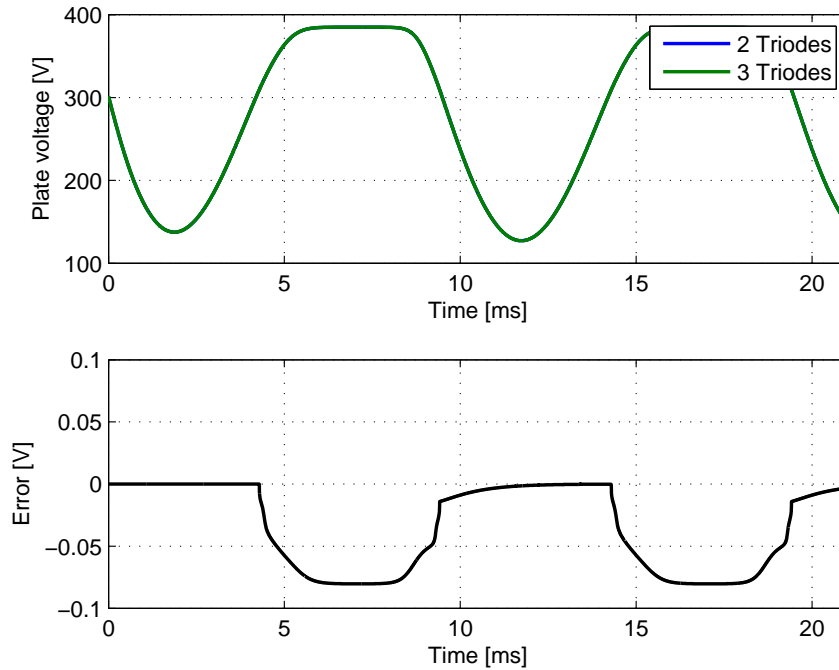


Figure 5.7: Comparison of the output signals of the first tube stage with one and two tube stages connected in series as the nonlinear load.

## 5.2 Guitar Tube Amplifier Simulation as a Case Study for Modified Block-wise Method

The previous chapter dealt with the general usage of the modified block-wise method for the simulation of the guitar preamp. The example of the whole guitar preamp simulation will be given in this chapter. The guitar preamp consisting of four tube amplifier stages was chosen as a case study. The circuit schematic of this preamp is shown in Figure 5.8 and values of circuit components are given by Table 5.2. This preamp is not a real preamp available on the market but it has the same topology and further, real preamps have slightly different values of components and they can contain capacitors connected between the cathode and ground in different tube stages not only in the first as in the schematic in Figure 5.8 [87]. Real preamps further contain the potentiometer for the regulation of the preamp gain. This potentiometer can be modeled by the resistors  $R_2$  and  $R_{g2}$  where the grid node of the second triode is connected to the tap of the potentiometer. Real preamps also contain the parametric passive filter, which is commonly called the tone stack. This circuit is mostly connected to the output of the preamp and therefore it is omitted in this simulation because it is the linear circuit.

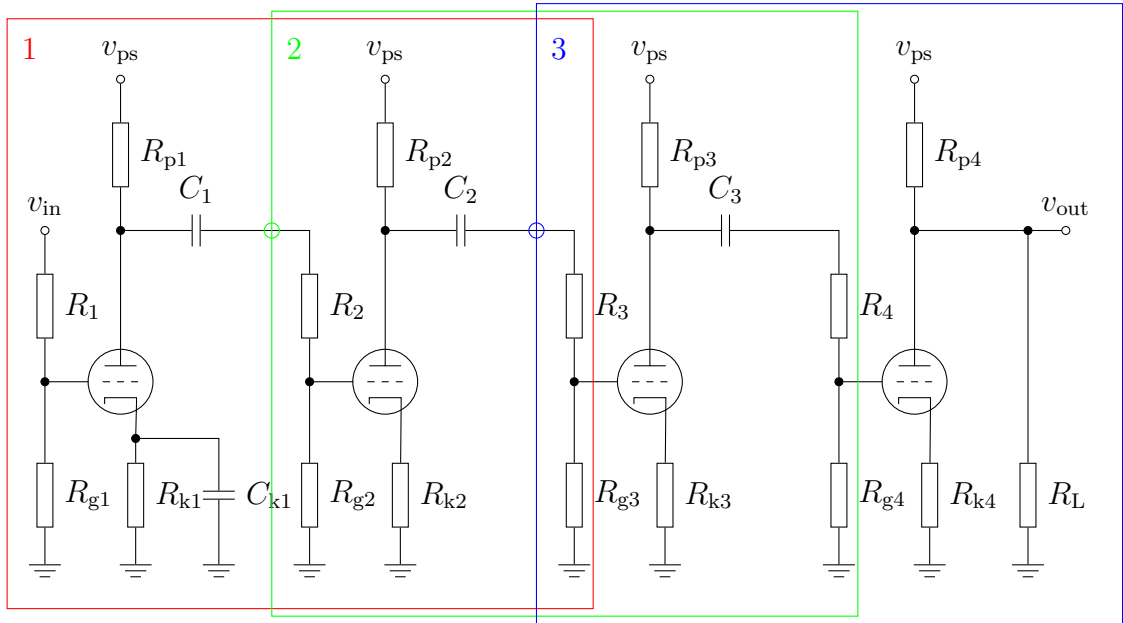


Figure 5.8: Circuit schematic for the guitar preamp with four tube stages and its decomposition into three blocks.

Table 5.2: Values for circuit components for the guitar preamp.

Component	Value
$R_1$	68 k $\Omega$
$R_{g1}, R_{g2}, R_{g3}, R_{g4}$	1 M $\Omega$
$R_{p1}, R_{p2}, R_{p3}, R_{p4}$	100 k $\Omega$
$R_{k1}$	2.7 k $\Omega$
$R_{k2}, R_{k3}, R_{k4}$	1.8 k $\Omega$
$R_2, R_3, R_4$	470 k $\Omega$
$R_L$	4 M $\Omega$
$C_1, C_2, C_3$	22 nF
$C_{k1}$	1 $\mu$ F
$V_{ps}$	400 V

Figure 5.8 also shows the decomposition into three separate blocks which is done in correspondence with the block diagram in Figure 5.1. Output and input signal nodes of the blocks are marked with green and blue circles. Green color marks the output node of the first block and the input of the second block and similarly blue color marks the output node of the second block and input node of the third block. Two types of blocks can be found after the decomposition and circuit schematics of both blocks are shown in Figures 5.9 and 5.10. Both types of blocks can be described in terms of the classic nodal analysis or the DK-method. Because the circuits are simple and contain at most two capacitors, the approximation of the direct ODEs is chosen, which will provide very efficient simulation algorithm. However, when the realistic behavior is desired rather than the minimal computational cost, the parasitic effect of the Miller capacitance and the power supply voltage fluctuation should be considered and in that case, the approximation of the DK-method would be more efficient.

The application of the nodal analysis to the first block (Figure 5.9) leads to the circuit equations

$$\begin{aligned}
0 &= (v_{\text{in}} - v_{\text{g1}})G_1 - v_{\text{g1}}G_{\text{g1}} - i_{\text{g1}} \\
0 &= v_{\text{c1m}} - v_{\text{c1}} - \frac{v_{\text{k}}G_{\text{k}} - i_{\text{p1}} - i_{\text{g1}}}{C_1f_s} \\
0 &= (v_{\text{ss}} - v_{\text{p1}})G_{\text{p1}} - (v_2 - v_{\text{g2}})G_2 - i_{\text{p1}} \\
0 &= v_{\text{c2m}} - v_{\text{c2}} - \frac{(v_2 - v_{\text{g2}})G_2}{C_2f_s} \\
0 &= (v_2 - v_{\text{g2}})G_2 - v_{\text{g2}}G_{\text{g2}} - i_{\text{g2}} \\
0 &= v_{\text{k2}}G_{\text{k2}} - i_{\text{g2}} - i_{\text{p2}} \\
0 &= (v_{\text{ss}} - v_{\text{p2}})G_{\text{p2}} - v_{\text{p2}}G_{\text{L}} - i_{\text{p2}}
\end{aligned} \tag{5.1}$$

where the capacitor currents were discretized using the BE discretization. The symbols  $G$  again denote conductances,  $f_s$  is the sampling frequency and functions  $i_{\text{g1}}$ ,  $i_{\text{p1}}$ ,  $i_{\text{g2}}$  and  $i_{\text{p2}}$  are nonlinear triode models discussed in the chapter 1.4.4. The second and the third block (see Figure 5.10) can be described by the equations

$$\begin{aligned}
0 &= (v_{\text{in}} - v_{\text{g1}})G_1 - v_{\text{g1}}G_{\text{g1}} - i_{\text{g1}} \\
0 &= v_{\text{k}}G_{\text{k}} - i_{\text{g}} - i_{\text{p}} \\
0 &= (v_{\text{ss}} - v_{\text{p1}})G_{\text{p1}} - (v_2 - v_{\text{g2}})G_2 - i_{\text{p1}} \\
0 &= v_{\text{c2m}} - v_{\text{c2}} - \frac{(v_2 - v_{\text{g2}})G_2}{C_2f_s} \\
0 &= (v_2 - v_{\text{g2}})G_2 - v_{\text{g2}}G_{\text{g2}} - i_{\text{g2}} \\
0 &= v_{\text{k2}}G_{\text{k2}} - i_{\text{g2}} - i_{\text{p2}} \\
0 &= (v_{\text{ss}} - v_{\text{p2}})G_{\text{p2}} - v_{\text{p2}}G_{\text{L}} - i_{\text{p2}}.
\end{aligned} \tag{5.2}$$



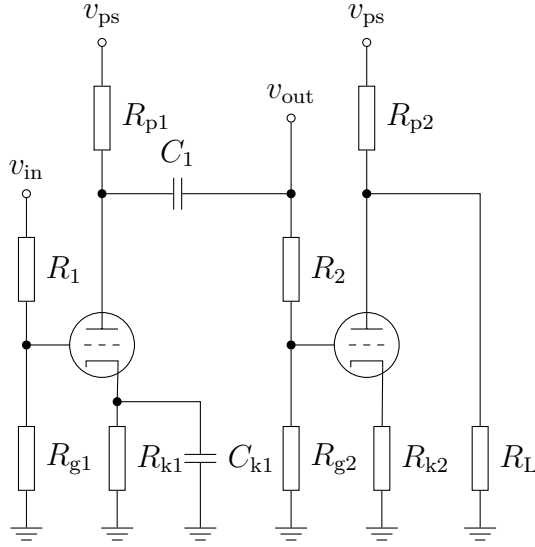


Figure 5.9: Circuit schematic of the block 1 of the decomposed tube preamp.

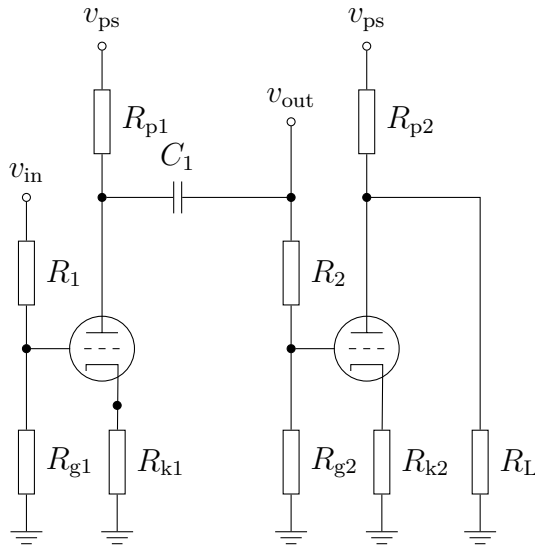


Figure 5.10: Circuit schematic of the block 1 and 2 of the decomposed tube preamp.

Both circuit equations (5.1) and (5.2) were numerically solved with the Koren's model of the triode 12AX7 with the generic model parameters given according to [70] for different input voltages and capacitor voltages. Approximating functions, the output function  $f_{out}$ , the capacitor  $C_1$  update function  $f_{c1}$  and the capacitor  $C_2$  update function  $f_{c2}$  defined in the chapter 4.1.1, for both blocks are plotted in Figures 5.11 and 5.12. Functions required by the first block are shown for the input voltage signal between -20 and 50 V and for capacitor voltages  $v_{C1} = 0$  V, 5 V and  $v_{C2} = 100$  V, 200 V, 300 V. Functions of the second block are plotted for the same input voltage and capacitor voltages  $v_{C1} = 0$  V, 100 V, 200 V, 300 V, 400 V.

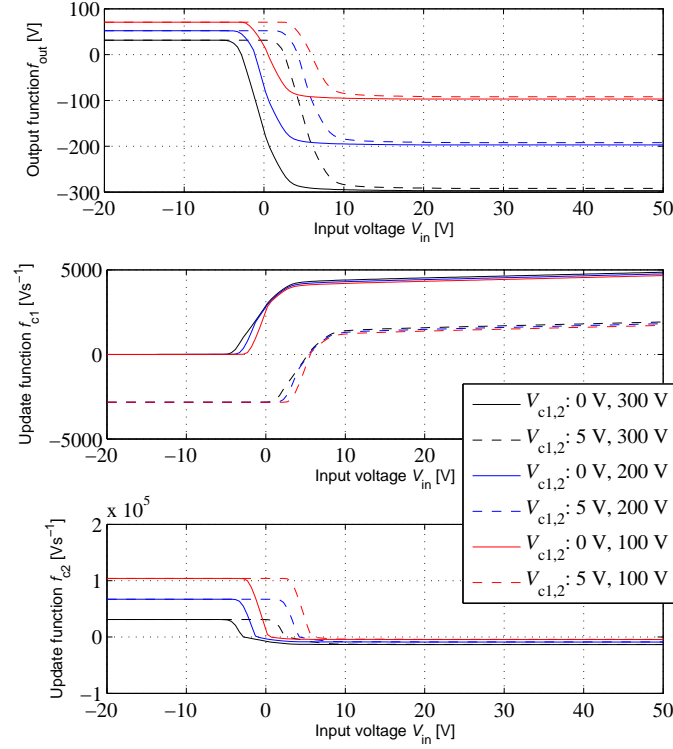


Figure 5.11: Approximating functions for the simulation of the system (5.1)

The individual functions have very similar shapes with different capacitor voltages. Therefore, the combination of the spline and the linear interpolation discussed in the chapter 4.5 can be used. Splines are used for the interpolation of the function for the different input voltage and constant capacitor voltages – the computed spline coefficients are stored in the look-up table in the row  $r$  which was computed as the linear function of the input and capacitor voltage indexes. The final function values are computed using the linear interpolation between function values for given capacitor values computed from spline coefficients stored in the look-up table. The step of input variables (the input voltage and capacitor voltages) was found experimentally. The maximal chosen deviation between the numerical solution and the approximation was 0.1 V. The input capacitor voltage  $v_{C1}$  has only two values because the technique of reshaping of the transfer function can be used. The step of the capacitor voltage  $v_{C2}$  was 5 V between 0 V and the power supply voltage 400 V. Other capacitor voltages are interpolated. The input voltage grid was  $\pm 200$  V with the step of 1 V. Details about the look-up table of the first block are in Table 5.3. The size of the look-up table is given by the number of the rows  $r$  of the table which

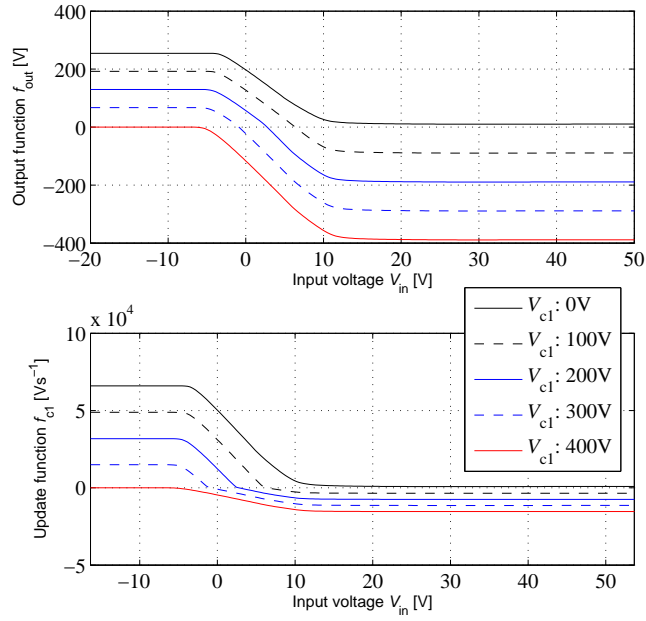


Figure 5.12: Approximating functions for the simulation of the system (5.1).

can be obtained from

$$n_r = v_{in\text{steps}} v_{c1\text{steps}} v_{c2\text{steps}} \quad (5.3)$$

and in case of the first block, it is 64000 rows. The size is 2 MB per table, if double precision floating point numbers are used. The total size of all tables (the output value function, the capacitor  $C_1$  update function and the capacitor  $C_2$  update function) is 6MB. The input voltage grid for the second and third block was  $\pm 200V$  with the step of 1V and the capacitor voltage  $v_{c1}$  grid was between 0 V and the power supply voltage with the step of 5V (see Table 5.4). The total size of both tables (the output value function, the capacitor  $C_1$  update function) is 2MB for one block. The total size of all tables required for the simulation of the whole preamp is 10 MB.

Table 5.3: Look-up table for simulation of the first block.

Variable	Min.	Max.	Step
$v_{in}[V]$	-200	200	1
$v_{c1}[V]$	0	5	5
$v_{c2}[V]$	0	400	5

Table 5.4: Look-up table for simulation of the second and third block.

Variable	Min.	Max.	Step
$v_{\text{in}}[V]$	-200	200	1
$v_{\text{c1}}[V]$	0	400	5

The final equation for the simulation of the first block are

$$\begin{aligned}
 v_{\text{out}}[n] &= f_{\text{out}}(v_{\text{in}}[n], v_{\text{c1}}[n], v_{\text{c2}}[n]) \\
 v_{\text{c1}}[n+1] &= v_{\text{c1}}[n] + T f_{\text{c1}}(v_{\text{in}}[n], v_{\text{c1}}[n], v_{\text{c2}}[n]) \\
 v_{\text{c2}}[n+1] &= v_{\text{c2}}[n] + T f_{\text{c2}}(v_{\text{in}}[n], v_{\text{c1}}[n], v_{\text{c2}}[n]).
 \end{aligned} \tag{5.4}$$

where  $n$  denotes the time index,  $v_{\text{out}}[n]$  is the output signal value of the first block and  $v_{\text{c1}}[n+1]$ ,  $v_{\text{c2}}[n+1]$  are capacitor voltages for the next sampling period. The equations of the second and third block are

$$\begin{aligned}
 v_{\text{out}}[n] &= f_{\text{out}}(v_{\text{in}}[n], v_{\text{c1}}[n]) \\
 v_{\text{c1}}[n+1] &= v_{\text{c1}}[n] + T f_{\text{c1}}(v_{\text{in}}[n], v_{\text{c1}}[n]).
 \end{aligned} \tag{5.5}$$

The final simulation equations (5.4),(5.5) are quite simple. This is the biggest advantage when comparing to other methods for the real time simulation, e.g. the DK-method, which requires matrix operations and the dimension of look-up tables would also have to be higher.

### 5.2.1 Computational Complexity

Because this thesis is focused on the real-time simulation of nonlinear audio effects, the computational complexity of designed algorithms plays very important role. First of all, the computational complexity of the numerical solution of the set of nonlinear functions  $\mathbf{f}(\mathbf{x}^i[n])$  describing the simulated system will be investigated. Generally, it is solved using the Newton-Raphson method

$$\mathbf{x}^{i+1}[n] = \mathbf{x}^i[n] - \mathbf{J}^{-1}(\mathbf{x}^i[n])\mathbf{f}(\mathbf{x}^i[n]) \tag{5.6}$$

where  $\mathbf{x}^i[n]$  is the vector of unknown variables,  $\mathbf{J}(\mathbf{x}^i[n])$  is the Jacobian matrix and  $\mathbf{f}(\mathbf{x}^i[n])$  is the function vector,  $i$  denotes the iteration index and  $n$  is the time index. Each iteration of the Newton-Raphson method requires the computation of the nonlinear function vector  $\mathbf{f}(\mathbf{x}^i[n])$  and the Jacobian  $\mathbf{J}(\mathbf{x}^i[n])$ . The computational cost of the vector  $\mathbf{f}(\mathbf{x}^i[n])$  depends on the particular simulated system and for now it can be expressed as  $c_f$ .

The Jacobian matrix  $\mathbf{J}(\mathbf{x}^i[n])$  contains partial first order derivatives of the function  $\mathbf{f}(\mathbf{x}^i[n])$ . Since the function  $\mathbf{f}(\mathbf{x}^i[n])$  does not have to be a continuous function, derivatives are generally computed using for instance the finite difference formula

$$f'(x) = \frac{1}{h} (f(x+h) - f(x)) \quad (5.7)$$

with the step  $h$  which consists of two function calls  $\mathbf{f}(\mathbf{x}^i[n])$  and two add operations and one multiply operation. The total cost of the Jacobian matrix computation is  $N+1$  function calls resulting in  $(c_f+3)N$  operations where  $N$  is the number of unknown variables. When the Jacobian matrix is established, its inversion matrix is computed. The computational complexity depends on the chosen algorithm of the matrix inversion. Generally, it is a  $O(N^3)$  problem. However, the LU decomposition offers more efficient implementation. Then, the equation (5.6) is rewritten as

$$\mathbf{LU} = \mathbf{J}(\mathbf{x}^i[n]) \quad (5.8)$$

$$\mathbf{Ly} = \mathbf{f}(\mathbf{x}^i[n]) \quad (5.9)$$

$$\mathbf{U}\Delta\mathbf{x}^i[n] = \mathbf{y} \quad (5.10)$$

$$\mathbf{x}^{i+1}[n] = \mathbf{x}^i[n] - \Delta\mathbf{x}^i[n]. \quad (5.11)$$

According to [91], the cost of the equation (5.8) (Crout's algorithm) is  $\frac{1}{3}N^3$  of inner loops containing one multiply and add operation, the cost of equations (5.9) and (5.10) is  $N^2$  multiply and add operations. Thus, the total cost is  $c_{\text{LU}} = \frac{2}{3}N^3 + 4N^2$  operations. Knowing  $\Delta\mathbf{x}^i[n]$ , the equation (5.11) can be solved, which requires  $N$  add operations. The total cost of the Newton method is then

$$c_{\text{nm}} = i \left( (c_f+3)N + c_f + \frac{2}{3}N^3 + 4N^2 + N \right) \quad (5.12)$$

where  $i$  is the number of iterations of the Newton method and  $N$  is the number of circuit nodes or unknown variables. However, it must be said that this number is theoretical. Neither the algorithm branches nor memory movements have been considered.

To compare the computational cost of the preamp simulation using the block division and without the division, the function cost  $c_f$  have to be determined as well as the number of unknown variables. The cost  $c_f$  can be obtained from (5.1) and (5.2) for both types of blocks and the function  $\mathbf{f}(\mathbf{x}^i[n])$  required for the whole preamp is given in [11]. All functions suppose the implementation of nonlinear tube models using look-up tables with the linear interpolation. As a result, the cost  $c_g$  of the grid current function is one add operation and two multiply operations, the cost  $c_p$  of the plate current function is three add operations and six multiply operations. The cost  $c_f$  of the whole preamp is  $c_f = 78 + 4c_p + 4c_g$  operations and  $N = 15$ .

The function  $\mathbf{f}(\mathbf{x}[n])$  of the first block involves 15 multiply operations and 22 add operations and four nonlinear function calls resulting in  $c_f = 37 + 2c_p + 2c_g$  operations. The equation is solved using the Newton method as well. Therefore, the equation (5.12) can be used for the determination of the computational complexity. For given  $N = 7$ , the total cost is

$$c_{\text{nm}} = i(749 + 16(c_g + c_p)). \quad (5.13)$$

The function  $\mathbf{f}(\mathbf{x}[n])$  of the second block has  $c_f = 34 + 2c_p + 2c_g$  operations and the cost is then

$$c_{\text{nm}} = i(728 + 16(c_g + c_p)). \quad (5.14)$$

If the blocks are connected together, the total cost of the solution of these blocks is

$$\begin{aligned} c_{\text{nm}} = & i_1(749 + 16(c_g + c_p)) + \\ & + (i_2 + i_3)(728 + 16(c_g + c_p)) \end{aligned} \quad (5.15)$$

where  $i_1$ ,  $i_2$  and  $i_3$  are numbers of iterations of the first, second and third block respectively.

Table 5.5 shows the number of operations required for the simulations based on the Newton method. Since the Newton method is the iterative process, the number of operations was investigated for one iteration, for the average number of iterations and the maximal number of iterations per sample as well. However, the average and the maximal number of iterations depend on the type of the input signal. Therefore, the algorithms were tested with an E-chord guitar riff with the maximal amplitude around 200 mV. The whole circuit simulation required 2.79 iterations on average and 100 iterations at the most. In case of the simulation using the block decomposition, average numbers of iterations were  $i_1 = 2.1294$ ,  $i_2 = 2.4149$ ,  $i_3 = 2.6806$  and maximal numbers of iterations were  $i_1 = 4$ ,  $i_2 = 5$ ,  $i_3 = 100$  for each block respectively. The number of iterations of individual blocks differs because each block processes different signals. The maximal number of iterations was 100 and the error limit of the Newton method was  $1 \times 10^{-5}$ . The number of iterations was computed from the whole signal (5 s,  $240 \times 10^3$  samples).

The computational complexity of algorithms based on different types of approximations discussed in the chapter 4 is shown in Table 5.6. There are available results for the whole preamp simulation as well as for the block-wise simulation. The numbers were computed from equations (5.4) and (5.5) where different types of approximation of the appropriate order  $N$  from the section 4 were used. Similarly, the whole system can be approximated by the order of approximation  $N = 5$ , because the whole circuit contains 4 capacitors.

Table 5.5: Computational complexity comparison of the simulations based on the Newton method – the number of operations.

Simulation type	One Iteration	Maximal Iteration	Average Iteration
Whole	$5.48 \times 10^3$	$5.48 \times 10^5$	$1.53 \times 10^4$
By blocks	$2.97 \times 10^3$	$1.39 \times 10^4$	$6.89 \times 10^3$

Table 5.6: Computational complexity comparison of the simulations based on approximations – the number of operations.

Simulation type	Linear Interp.	Cubic Interp.	Spline Interp.
Whole	$9.38 \times 10^2$	$5.28 \times 10^4$	$9.53 \times 10^2$
By blocks	$2.06 \times 10^2$	$2.58 \times 10^3$	$2.51 \times 10^2$

As the results available in Tables 5.5 and 5.6 have shown, algorithms based on approximations offer the constant computational complexity, which is also much lower than at the algorithms based on the Newton method.

The algorithm with the spline interpolation and the block-wise method was implemented in C++ language as the Virtual Studio Technology (VST) plug-in effect to test it in real-time. It was tested with a 2.66 GHz i7 Intel Mac with 4 GB RAM at the sampling frequency of 48 kHz using the external audio interface M-Audio Fast Track Pro with the Audio Stream Input/Output (ASIO) buffer size 128 samples. If no oversampling was used, the CPU load was around 3 %.

## 5.2.2 Simulation Results

The proposed algorithm was then tested with different input signals including sine wave signals at different frequencies and amplitudes (including frequencies close to the Nyquist frequency), a logarithmic sweep signal (see Figure 5.13) and also the real guitar signal. All the performed simulations were stable even if no oversampling was used and the amplitude of the testing signal was around hundreds of volts at the input of the third block. The comparison between the simulation of the whole preamp using the Newton method and the simulation based on the division into blocks is plotted in Figure 5.14. Time difference signals are normalized to the maximal value of the output signal.

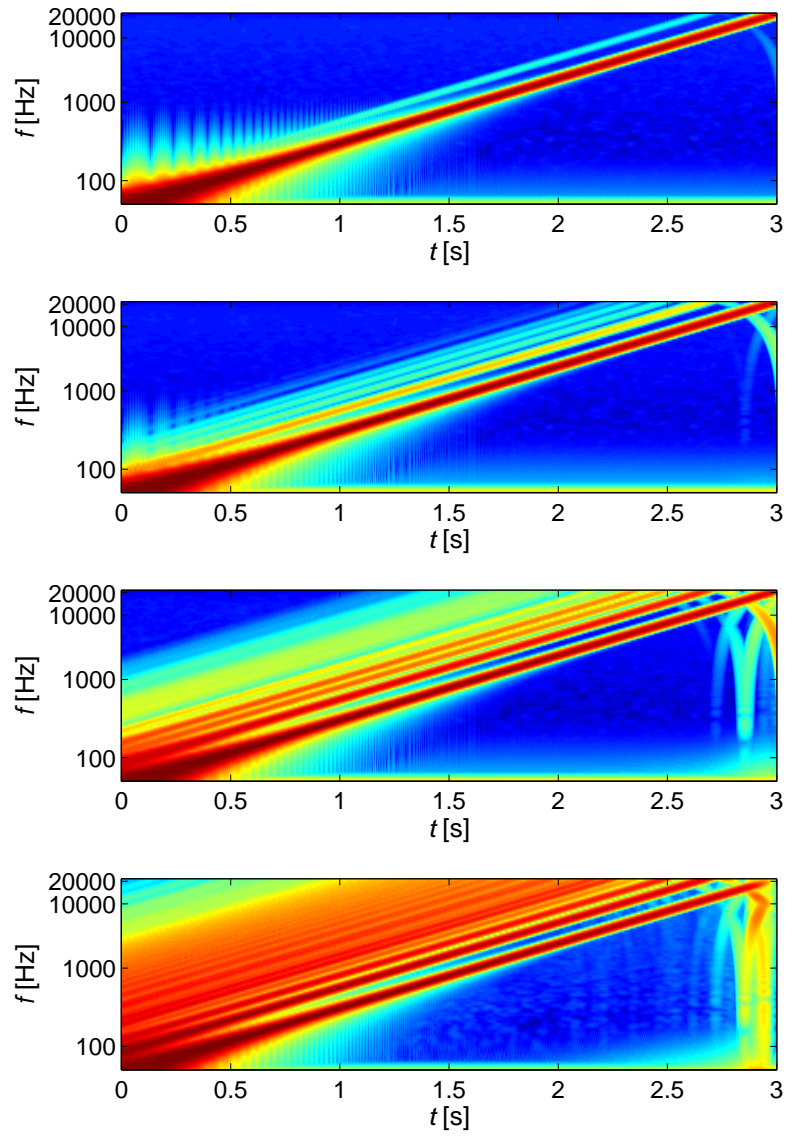


Figure 5.13: Simulation results for a logarithmic sweep signal. The plate voltage signals of all tubes are displayed.



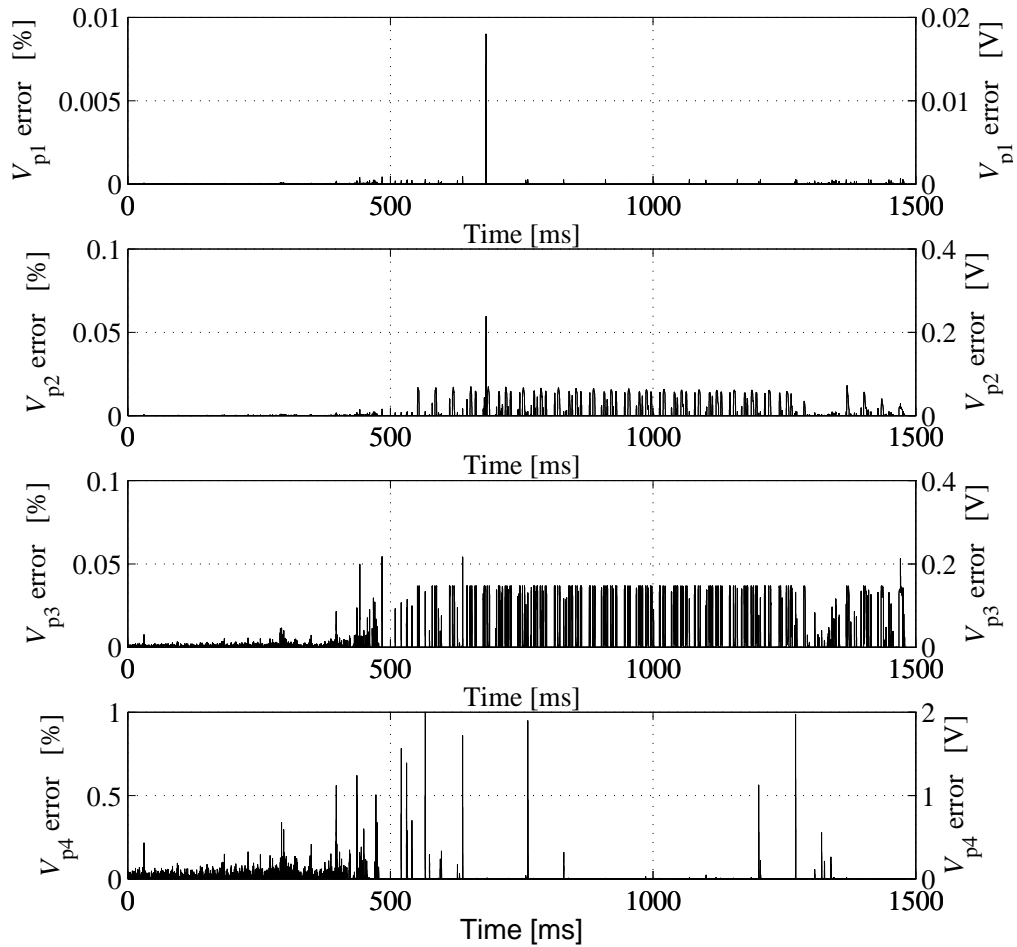


Figure 5.14: Comparison between simulation results using the numerical solution and using approximations for a part of the real guitar riff. Only the error signals are displayed.

Error values, such as the maximal and the average error, are also expressed in Table 5.7. The maximal error is around 2 V. The amplification of the preamp is approximately  $1.84 \times 10^4$  and therefore it is probable that this error can be masked by the amplified noise of the connected guitar and by the noise of the preamp. In places with the maximal error, the Newton method reached the maximal number of iterations (100), and therefore the error is caused partly by the approximation and partly by the Newton method.

Table 5.7: Errors for the simulation from Figure 5.14. The plate voltage signal errors are displayed.

–	$v_{p1}$	$v_{p2}$	$v_{p3}$	$v_{p4}$
Max[V]	$6.27 \times 10^{-4}$	$2.38 \times 10^{-1}$	$2.12 \times 10^{-1}$	1.99
Mean[V]	$1.13 \times 10^{-5}$	$5.70 \times 10^{-3}$	$1.01 \times 10^{-2}$	$6.20 \times 10^{-3}$
Var[V]	$1.05 \times 10^{-7}$	$2.54 \times 10^{-4}$	$1.07 \times 10^{-3}$	$0.13 \times 10^{-3}$

### 5.3 Decomposition of the DK-method nonlinear core

The block decomposition discussed in the previous section provides very efficient simulation of circuits where the number of circuit inputs represented by real inputs and energy-storing components is lower than the number of nonlinear equations. Otherwise, the DK-method is more appropriate and it can be of course used together with the modified block-wise method. The advantage of this approach is the simulation of the circuit with more energy-storing components. This approach will be studied on the simulation of the guitar preamp with topology typical for Fender preamps [87]. This chapter was published mainly in [12]. The circuit schematic of the studied preamp is in Figure 5.15 with component values in Table 5.8 [95]. It consists of two triodes 12AX7 and the tone stack connected between them. Components  $R_6$ ,  $R_7$ ,  $R_8$  and  $R_9$  are potentiometers of variable circuit parameters treble  $\theta$ , middle  $\mu$ , bass  $\beta$  and volume  $\lambda$  with the following mapping to resistors:  $R_{6b} = R_6\theta$ ,  $R_{6a} = R_6(1 - \theta)$ ,  $R_7\beta$ ,  $R_8\mu$   $R_{9b} = R_9\lambda$  and  $R_{9a} = R_9(1 - \lambda)$ .

Table 5.8: Component values of the Fender type guitar preamp

Component	Value	Component	Value	Component	Value
$R_1$	32 k $\Omega$	$R_2$	1 M $\Omega$	$R_3$	1.5 k $\Omega$
$R_4$	100 k $\Omega$	$R_5, R_{11}$	100 k $\Omega$	$R_6$	250 k $\Omega$
$R_7$	250 k $\Omega$	$R_8$	10 k $\Omega$	$R_9$	1 M $\Omega$
$R_{10}$	1.64 k $\Omega$	$C_1, C_5$	22 $\mu$ F	$C_2$	250 pF
$C_3$	100 nF	$C_4$	22 nF	$V_{ps}$	385 V

DK-method matrices can be computed from incidence matrices which are given in the appendix C. These matrices differ from the original matrices used in [12] because the efficient recomputation of DK-method matrices, as was discussed in the chapter 1.3.2 and used for this preamp simulation in [12], is not considered

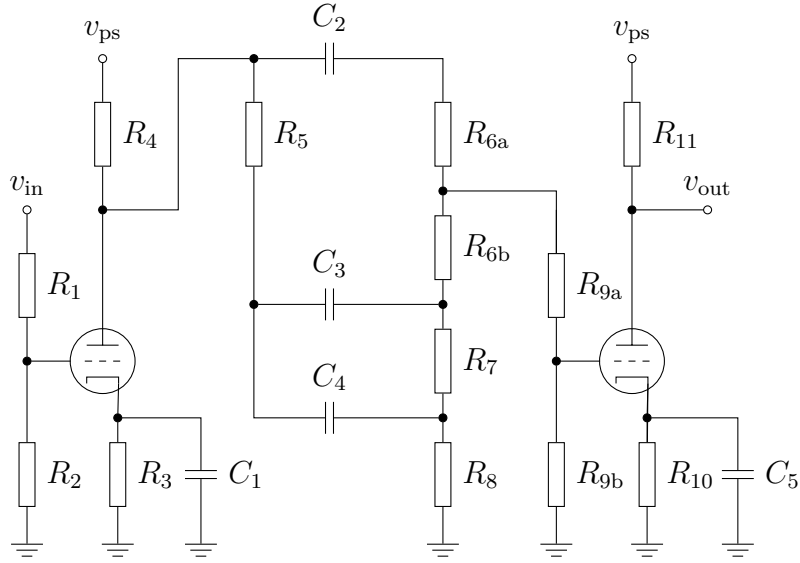


Figure 5.15: Circuit schematic of the Fender type guitar preamp.

because the main focus of this chapter is on the efficient computation of the nonlinear core of the DK-method. Using equations (1.62) – (1.70) the following matrices are derived:  $\mathbf{A} \in \mathbb{R}^{5 \times 5}$ ,  $\mathbf{B} \in \mathbb{R}^{5 \times 2}$ ,  $\mathbf{C} \in \mathbb{R}^{5 \times 4}$ ,  $\mathbf{D} \in \mathbb{R}^{1 \times 5}$ ,  $\mathbf{E} \in \mathbb{R}^{1 \times 2}$ ,  $\mathbf{F} \in \mathbb{R}^{1 \times 4}$ ,  $\mathbf{G} \in \mathbb{R}^{4 \times 5}$ ,  $\mathbf{H} \in \mathbb{R}^{4 \times 2}$ ,  $\mathbf{K} \in \mathbb{R}^{4 \times 4}$  and the nonlinear function to be solved is

$$\mathbf{0} = \begin{pmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{pmatrix} + \mathbf{K} \begin{pmatrix} i_g(v_{gk1}, v_{pk1}) \\ i_p(v_{gk1}, v_{pk1}) \\ i_g(v_{gk2}, v_{pk2}) \\ i_p(v_{gk2}, v_{pk2}) \end{pmatrix} - \begin{pmatrix} v_{gk1} \\ v_{pk1} \\ v_{gk2} \\ v_{pk2} \end{pmatrix}. \quad (5.16)$$

### 5.3.1 Precomputation

As it can be seen from (5.16) and from the dimension of the matrix  $\mathbf{K}$ , the nonlinear equation has 4 independent inputs for which it has to be precomputed. First of all, it is necessary to find ranges for input variables  $p_1$ ,  $p_2$ ,  $p_3$  and  $p_4$ . This can be done using  $\mathbf{p}[n] = \mathbf{G}\mathbf{x}[n] + \mathbf{H}\mathbf{u}[n]$  and it will depend on input signal values and circuit state values as well. The range of input values can be easily derived from input signal properties and also from the power supply value  $V_{PS}$ , obtaining the range of state variables is much more complicated. Values of state variables will belong to the interval  $[0, V_{PS}]$  but mostly, the range will be much narrower. However, the ranges can also be estimated from performed simulations without the approximation. Ranges of input variables  $p_1$ ,  $p_2$ ,  $p_3$  and  $p_4$  used for this simulation are in Table 5.9.

Because the circuit contains variable circuit components, the coefficients of the  $\mathbf{K}$  matrix are not constant. Ranges for  $\mathbf{K}$  matrix coefficients can be obtained from (1.70) for variable conductances dependent on parameter values  $\theta$ ,  $\beta$ ,  $\mu$ , and  $\lambda$ . In

this case, the derivation of parameter ranges is much easier, the range is  $[0, 1]$  for all parameters and the range of  $\mathbf{K}$  matrix coefficients is given by the minimal and the maximal value for all combinations of discretized parameter values. The ranges of  $\mathbf{K}$  coefficients are in Table 5.10.

Table 5.9: Ranges of  $\mathbf{p}$  parameters.

	$p_1$	$p_2$	$p_3$	$p_4$
min	-4	200	-400	200
max	4	400	400	400
step	0.125	4	0.25	0.25

Table 5.10: Ranges of  $\mathbf{K}$  coefficients.

	$k_{22}$	$k_{23}$	$k_{33}$
min	$3.3 \times 10^4$	0.0	0.4
max	$4.6 \times 10^4$	$3.8 \times 10^4$	$1.3 \times 10^5$

As the result of the precomputation, there are four 4D look-up tables for circuit currents or voltages for constant parameters or 7D look-up tables for the parametric circuit. The  $p_1$ ,  $p_2$ ,  $p_3$  and  $p_4$  parts of look-up tables should be interpolated with a smooth interpolation, while the linear interpolation is sufficient for the parametric part. The process of the precomputation itself is rather computationally demanding. Because the multivariate nonlinear equation tends to oscillate, one must use very small step between neighboring data points to force the solution to converge and it costs a lot of time and memory. After the precomputation, it is needed 53 GB of memory for coefficients per table. Although the number of data points can be reduced by using the nonuniform approximation, algorithms for the nonuniform reduction of data discussed in the chapter 4 needs lot of time to process all the data and therefore the straightforward approximation is not very convenient.

### 5.3.2 Further Look-up Table Size Reduction

Look-up tables designed in the previous chapter are sufficient for working in real-time. Nevertheless, they are quite impractical for the implementation of the algorithm which would simulate the circuit because the amount of data is still large. Therefore, a further compression of data would be beneficial. By closer look at the  $\mathbf{K}$  matrix, one can observe some zero or low-value coefficients compared to other values.

As a result of this, the correlation analysis of precomputed data was performed. Covariance matrices for input variables and function values were in the form (expressed only for one row of input data)

$$C = \text{cov} \left( \left[ p_1^{i_1} p_2^{i_2} p_3^{i_3} p_4^{i_4} f(p_1^{i_1}, p_2^{i_2}, p_3^{i_3}, p_4^{i_4}) \right] \right) \quad (5.17)$$

where the superscripts denote indexes  $i_1 \in [1, N_1]$ ,  $i_2 \in [1, N_2]$ ,  $i_3 \in [1, N_3]$  and  $i_4 \in [1, N_4]$  for the number of values  $N_1$ ,  $N_2$ ,  $N_3$  and  $N_4$  computed from ranges in Table 5.9 and function  $f(p_1, p_2, p_3, p_4)$  was substituted with precomputed nonlinear current functions  $i_{g1}(\mathbf{p})$ ,  $i_{p1}(\mathbf{p})$ ,  $i_{g2}(\mathbf{p})$  and  $i_{p2}(\mathbf{p})$ . The resulting covariance between nonlinear functions and inputs  $p_1$ ,  $p_2$ ,  $p_3$  and  $p_4$  is stated in Table 5.11.

Table 5.11: Covariance between precomputed functions and inputs.

	$i_{g1}(\mathbf{p})$	$i_{p1}(\mathbf{p})$	$i_{g2}(\mathbf{p})$	$i_{p2}(\mathbf{p})$
$p_1$	$-5.23 \cdot 10^{-4}$	$-6.98 \cdot 10^{-3}$	$5.55 \cdot 10^{-4}$	$8.61 \cdot 10^{-4}$
$p_2$	$1.12 \cdot 10^{-6}$	$-1.02 \cdot 10^{-1}$	$6.65 \cdot 10^{-3}$	$1.17 \cdot 10^{-2}$
$p_3$	$-1.17 \cdot 10^{-8}$	$1.11 \cdot 10^{-3}$	$-1.40 \cdot 10^{-2}$	$-2.47 \cdot 10^{-2}$
$p_4$	$5.49 \cdot 10^{-14}$	$5.02 \cdot 10^{-9}$	$5.40 \cdot 10^{-8}$	$-8.81 \cdot 10^{-3}$

As it can be seen from the table, some nonlinear functions are almost independent on some input variables – functions  $i_{g1}(\mathbf{p})$ ,  $i_{p1}(\mathbf{p})$ ,  $i_{g2}(\mathbf{p})$  are independent on the variable  $p_4$  and the function  $i_{g1}(\mathbf{p})$  is further almost independent on the variable  $p_3$ . Therefore, the look-up table  $i_{g1}(\mathbf{p})$  can have only 2 dimensions for constant parameters and 5 dimensions for the parametric circuit, the look-up tables  $i_{p1}(\mathbf{p})$ ,  $i_{g2}(\mathbf{p})$  are 3D for constant and 6D for the parametric circuit and the look-up table  $i_{p2}(\mathbf{p})$  remains the same.

The second important point is that due to missing connections between nonlinear functions, the further simplification and decomposition can be done. The nonlinear equation (5.16) can be split in this case into

$$\begin{aligned} v_{gk1} &= k_{11}i_g(v_{gk1}) + k_{12}i_p(v_{gk1}, v_{pk1}) + p_1 \\ v_{pk1} &= k_{21}i_g(v_{gk1}) + k_{22}i_p(v_{gk1}, v_{pk1}) + \underbrace{k_{23}i_g(v_{gk2}) + p_2}_{\overline{p_2}} \end{aligned} \quad (5.18)$$

and

$$\begin{aligned} v_{gk2} &= \underbrace{p_3 + k_{32}i_p(v_{gk1}, v_{pk1})}_{\overline{p_3}} + k_{33}i_g(v_{gk2}) + k_{34}i_p(v_{gk2}, v_{pk2}) \\ v_{pk2} &= k_{43}i_g(v_{gk2}) + k_{44}i_p(v_{gk2}, v_{pk2}) + p_4 \end{aligned} \quad (5.19)$$

where terms  $k_{23}i_g(v_{gk2})$  and  $k_{32}i_p(v_{gk1}, v_{pk1})$  are mutual impacts of adjacent triodes. Equations (5.18) and (5.19) can be computed separately for input variables  $p_1, \bar{p}_2$  and  $\bar{p}_3, p_4$  and further functions

$$\overline{f_{ip1}}(p_1, \bar{p}_2) = i_p(v_{gk1}(p_1, \bar{p}_2), v_{pk1}(p_1, \bar{p}_2)) \quad (5.20)$$

$$\overline{f_{ig2}}(\bar{p}_3, p_4) = i_g(v_{gk1}(\bar{p}_3, p_4), v_{pk1}(\bar{p}_3, p_4)) \quad (5.21)$$

can be introduced. Subsequently, the plate current  $i_{p1}$  can be computed from

$$i_{p1} = \overline{f_{ip1}}(p_1, p_2 + \overline{f_{ig2}}(p_3 + i_{p1}, p_4)) \quad (5.22)$$

for  $p_1, p_2, p_3$  and  $p_4$  then

$$i_{g2} = \overline{f_{ig2}}(p_3 + i_{p1}, p_4) \quad (5.23)$$

and finally, all remaining currents or voltages can be derived from (5.18) and (5.19).

Considering this decomposition together with the reduction of look-up tables based on the correlation analysis, the whole model is split into two independent parts, each containing one tube. Grid and plate currents are computed from the approximated functions

$$i_{g1} = i_{g1app}(p_1, p_2) \quad (5.24)$$

$$i_{p1} = i_{p1app}(p_1, p_2, p_3) \quad (5.25)$$

where the redundant inputs were neglected. Then, the mutual interaction of both tubes is known and it has been already included in the plate current  $i_{p1}$ . This current is subsequently used as the additional contribution to the input  $p_3$  as it can be seen in (5.19) and currents of the second tube are obtained from

$$i_{g2} = i_{g2app}(p_3 + k_{32}i_{p1}) \quad (5.26)$$

$$i_{p2} = i_{g2app}(p_3 + k_{32}i_{p1}, p_4). \quad (5.27)$$

The big advantage of this decomposition is that it should not introduce any error (results can differ but it is rather caused by the numerical solving) if no  $p$  parameter is omitted. The last task is to consider the efficient approximation regarding variable  $\mathbf{K}$  matrix coefficients –  $k_{22}$ ,  $k_{23} = k_{32}$  and  $k_{33}$ . The grid current  $i_{g1}$  is independent on the second tube, therefore it should only depend on the coefficient  $k_{22}$ , but in this case the current  $i_{g1}$  does not change with the value of  $k_{22}$ . The plate current  $i_{p1}$  depends on all variable  $k$  coefficients and currents of the second tube depend only on the coefficient  $k_{33}$  because other coefficients have already been included in the contribution of the plate current to the variable  $p_3$ . Eventually, it was found that using parameters  $\theta, \beta, \mu$ , and  $\lambda$  as inputs into the look-up table for the current  $i_{p1}$  gives the smaller look-up table size than using coefficients  $k_{22}, k_{23}, k_{33}$  although the

number of variable  $\mathbf{K}$  coefficients is smaller than the number of parameters. This is because the function  $i_{p1}$  changes dramatically with different  $k_{22}$ ,  $k_{23}$ ,  $k_{33}$  values and thus coefficients  $k_{22}$ ,  $k_{23}$ ,  $k_{33}$  must be more densely sampled, while with parameters  $\theta$ ,  $\beta$ ,  $\mu$ ,  $\lambda$  as the inputs into the look-up table, it seems to be sufficient to use only boundary values. Furthermore, the current  $i_{p1}$  is almost independent on the bass parameter  $\beta$  – this is due to the parallel combination of resistors  $R_{18}$  and  $R_{22}$  with the serial combination  $R_{20}\beta$  and  $R_{19}$ .

The final equations are

$$i_{g1} = i_{g1app}(p_1, p_2) \quad (5.28)$$

$$i_{p1} = i_{p1app}(p_1, p_2, p_3, \theta, \mu, \lambda) \quad (5.29)$$

$$i_{g2} = i_{g2app}(p_3 + k_{32}i_{p1}, k_{33}) \quad (5.30)$$

$$i_{p2} = i_{g2app}(p_3 + k_{32}i_{p1}, p_4, k_{33}) \quad (5.31)$$

and the detailed description of look-up tables is provided in Table 5.12 where the number of intervals per dimension per table is given. It consists of spline interpolations for  $\mathbf{p}$  parameters and linear interpolations for circuit parameter changes. During the processing of the input signal, it is not necessary to compute all interpolations, the parametric part can be interpolated only when parameters are changed and resulting interpolated spline coefficients are stored in the runtime memory of the algorithm. Then only the interpolation based on  $\mathbf{p}$  variables is performed for each signal sample. The size of the runtime memory is shown in the third column of Table 5.12. The memory size required for the look-up tables is mentioned in Table 5.13. Comparing to the original data size 53 GB per table (based on Table 5.9 and boundary values for  $\theta$ ,  $\beta$ ,  $\mu$ , and  $\lambda$ ), the great reduction of data has been achieved. However, it is not sufficient for some applications, e.g. implementations on signal processors. In this case, it is possible to perform the interpolation directly from data points stored in look-up tables instead of spline coefficients. Then, the look-up table size is reduced (see the columns "local interpolation" in Table 5.13) but on the other hand, the runtime computation of whole splines will significantly increase the overall computational complexity. In such case, it is better to use a local interpolation working on base of the nonuniform grid only from neighboring data points. However, these interpolations will be slower than the proposed spline interpolation (see chapter 4 for the comparison of different approximations).

Table 5.12: Description of the look-up tables – number of intervals

table	spline	linear interpolation	runtime
$i_{g1app}$	$13 \times 6$	-	$13 \times 6$
$i_{p1app}$	$46 \times 14 \times 29$	$2 \times 2 \times 2$	$46 \times 14 \times 29$
$i_{g2app}$	$15 \times 6 \times 13$	-	$15 \times 6$
$i_{p2app}$	$38 \times 8 \times 12$	-	$38 \times 8$

Table 5.13: Data size of the look-up tables.

table	global interpolation [kB]		local interpolation [kB]	
	total	runtime	total	runtime
$i_{g1app}$	3.75	3.75	0.30	0.30
$i_{p1app}$	65520.00	4095.00	1167.26	72.95
$i_{g2app}$	210.00	4.38	4.57	0.35
$i_{p2app}$	712.25	16.20	14.25	1.19

### 5.3.3 Simulation Results

The algorithm for the simulation of the preamp was written as the Matlab mex function in C language. The algorithm consists of nonuniform spline interpolations up to third order for the  $p$  variable interpolation and linear interpolations for the parameter variable interpolation. The parameter interpolation is performed before the main processing loop. The Template Numerical Toolkit<sup>1</sup> was used for matrix operations. The computational complexity of the algorithm was around 10 % on the 2.66 GHz Intel processor but more than half of it was spent on matrix operations. The original state-space model without approximations consumes around 76 %. The algorithm was not optimized using the parallel processing because it would require to write critical parts of the algorithm in assembly language or using intrinsic functions. As the result of this, the further reduction of the computational complexity is possible. The quality of the approximation is illustrated in Figures 5.16, 5.17, 5.18, and 5.19. The input signal was a 100 Hz sine wave signal with an amplitude of 0.5 V and the sampling frequency value of 48 kHz. Figures 5.16 and 5.17 show output signals in the time and frequency domain for the numerical solution and the approximated solution for parameter values  $\theta = 1$ ,  $\beta = 1$ ,  $\mu = 1$ ,  $\lambda = 1$ ; that means without the interpolation of parameters. The interpolation for parameter values  $\theta = 0.5$ ,  $\beta = 0.5$ ,  $\mu = 0.5$ ,  $\lambda = 0.5$ , which is the worst case, is shown in Figures 5.18 and

<sup>1</sup><http://math.nist.gov/tnt/>



5.19. Although the difference of the numerical and approximated solution is quite large in the time domain, the harmonic content is very similar. The sine wave signal mentioned earlier and the real guitar riff were used as input signals. There was only a very subtle difference for the sine wave signal and the version with parameters  $\theta = 0.5$ ,  $\beta = 0.5$ ,  $\mu = 0.5$ ,  $\lambda = 0.5$ .

The most difficult part of the approximation is choosing the maximal approximation error which is used for the nonuniform data reduction. In this case, the chosen error for the data reduction was  $1 \times 10^{-6}$  A. Because the triode current functions were approximated, the output signal error is equal to the chosen error multiplied by the plate resistor value multiplied with the amplification factor of the next triode. The next error source are the constant  $p$  parameters. This error can be however only evaluated by comparing of output signals with non-constant  $p$  parameters.

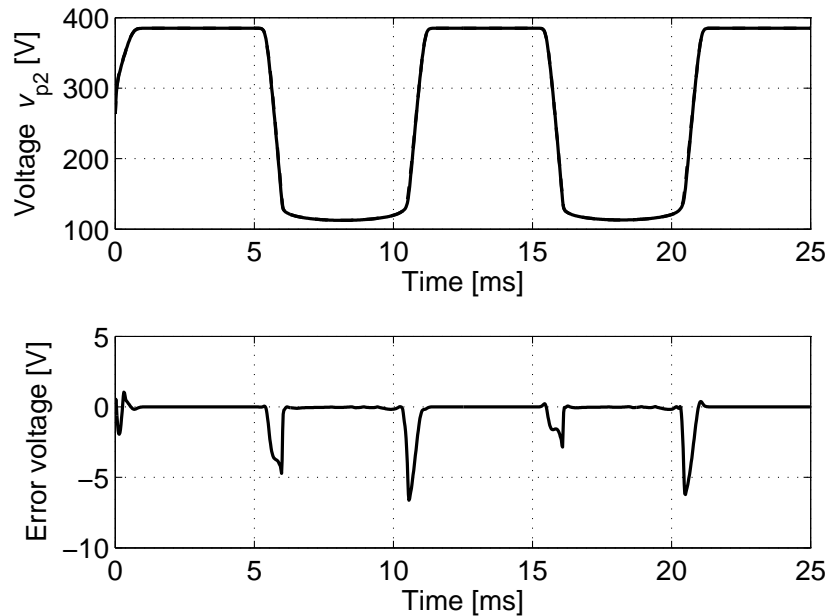


Figure 5.16: Output signals (top, dashed line for the numerical solution) and the difference between numerical and approximated solution in the time domain without the parameter interpolation.

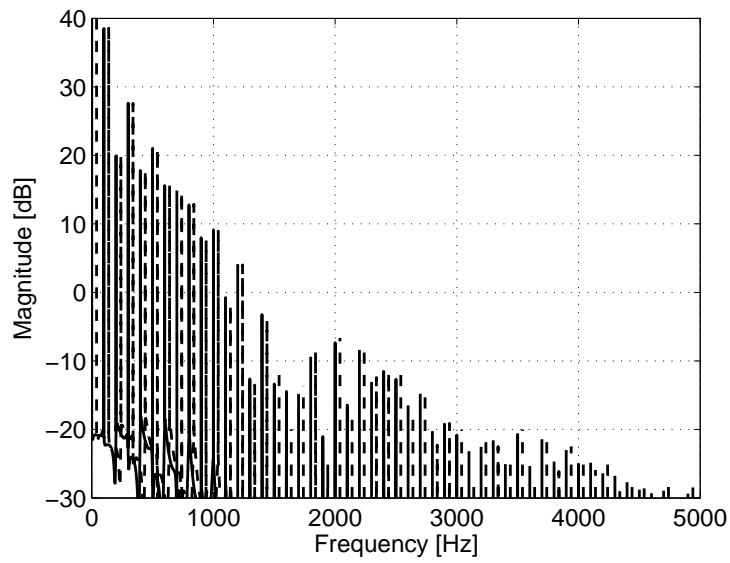


Figure 5.17: Difference between the numerical and approximated solution in the time frequency without the parameter interpolation. The numerical solution (dashed line) is shifted to the right.

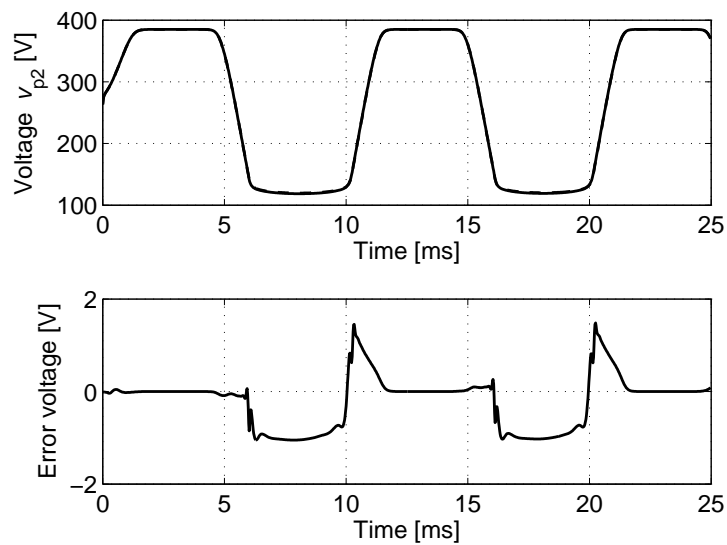


Figure 5.18: Output signals (top, dashed line for the numerical solution) and the difference between the numerical and approximated solution in the time domain with the parameter interpolation.

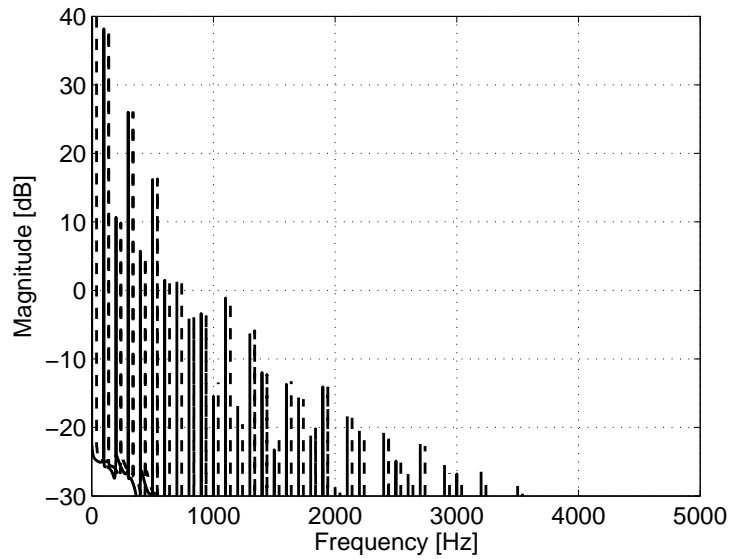


Figure 5.19: Difference between the numerical and approximated solution in the time frequency with the parameter interpolation. The numerical solution (dashed line) is shifted to the right.

## 5.4 DK-model Decomposition Using Connection Components

The decomposition described in the previous section allows the efficient decomposition of feed-forward circuits with triodes or transistors without any feedbacks. This is due to the form of the  $\mathbf{K}$  matrix, which is tri-diagonal, and the model of the triodes or transistors which are built by two nonlinear current functions controlled by two control voltages. Nevertheless, previous models of preamps did not take the Miller effect into account which in case of circuits with tubes can manifest in the audible area [65]. This effect is caused by the parasitic capacitance between the grid and the plate node of the tube and this effect can be simulated by connecting the parasitic capacitor into the model of the amplifier. This capacitance causes the local feedback of each tube amplifier stage between the plate and the grid circuit and as a result, the  $\mathbf{K}$  matrix does not have the tri-diagonal form anymore. This will be shown for the model of the Marshall JCM 800 preamp [87]. The circuit schematic of a part of this preamp (from the input of the preamp to the input of the cathode follower circuit which is connected to the node  $v_{\text{out}}$ ) is displayed in Figure 5.20 and component values in Table 5.14. Parasitic capacitors are marked with red color. The circuit contains one potentiometer for the "Gain" parameter which is modeled by resistors  $R_{6a}$  and  $R_{6b}$ .

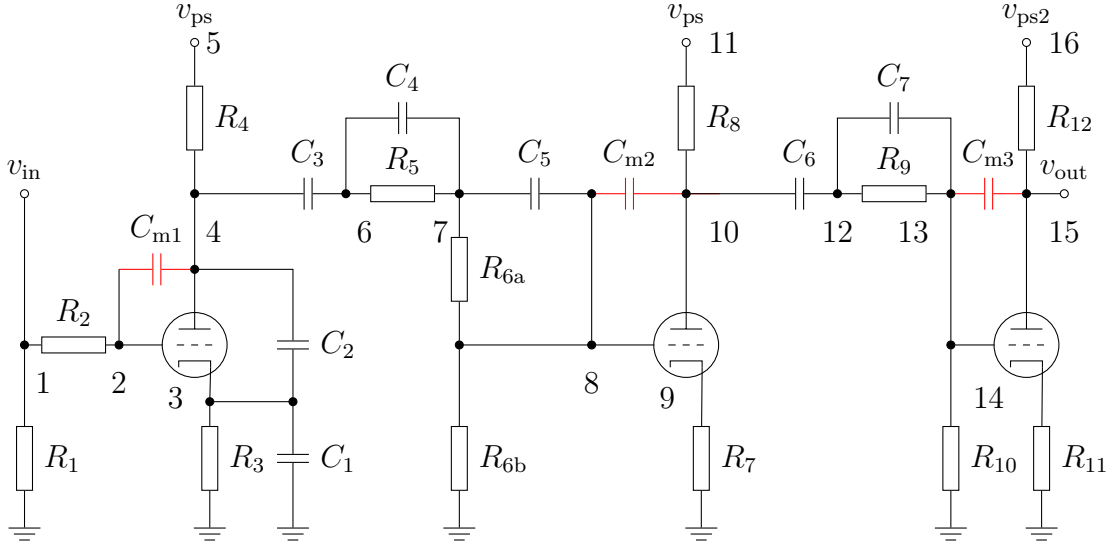


Figure 5.20: Circuit schematic of the Marshall JCM 800 guitar preamp.

Table 5.14: Component values of the Marshall JCM 800 guitar preamp

Component	Value	Component	Value	Component	Value
$R_1, R_6$	1 M $\Omega$	$R_2$	68 k $\Omega$	$R_3$	2.7 k $\Omega$
$R_4, R_8, R_{12}$	100 k $\Omega$	$R_5, R_9, R_{10}$	470 k $\Omega$	$R_7$	10 k $\Omega$
$R_{11}$	820 $\Omega$	$C_1$	680 nF	$C_2$	100 pF
$C_3, C_6$	22 nF	$C_4, C_7$	470 pF	$C_5$	1 nF
$C_{m1}, C_{m2}, C_{m3}$	2 pF	$V_{ps}$	385 V	$V_{ps2}$	385 V

DK-method matrices  $\mathbf{A} \in \mathbb{R}^{10 \times 10}$ ,  $\mathbf{B} \in \mathbb{R}^{10 \times 4}$ ,  $\mathbf{C} \in \mathbb{R}^{10 \times 6}$ ,  $\mathbf{D} \in \mathbb{R}^{1 \times 10}$ ,  $\mathbf{E} \in \mathbb{R}^{1 \times 4}$ ,  $\mathbf{F} \in \mathbb{R}^{1 \times 6}$ ,  $\mathbf{G} \in \mathbb{R}^{6 \times 10}$ ,  $\mathbf{H} \in \mathbb{R}^{6 \times 4}$ ,  $\mathbf{K} \in \mathbb{R}^{6 \times 6}$  can be computed using (1.62) – (1.70) from the incidence matrices  $\mathbf{N}_r$ ,  $\mathbf{N}_x$ ,  $\mathbf{N}_n$ ,  $\mathbf{N}_u$ ,  $\mathbf{N}_o$  and conductance matrices  $\mathbf{G}_r$  and  $\mathbf{G}_x$  which are given in the appendix D. Matrices without considering the parasitic capacitances can be computed from incidence matrices where the last three rows of the matrix  $\mathbf{N}_x$  are excluded as well as the last three members in the definition of the diagonal matrix  $\mathbf{G}_x$  or if capacitance values of the parasitic capacitors are set to zero. Both approaches give different matrices  $\mathbf{A}$ ,  $\mathbf{C}$  and  $\mathbf{G}$  but the matrix  $\mathbf{K}$  is the same, which is important for the analysis of the nonlinearity. The  $\mathbf{K}$  matrix without

considering the parasitic capacitances is

$$\mathbf{K} = \begin{bmatrix} 6.80 \times 10^4 & 8.13 & -6.94 & 0 & 0 & 0 \\ 8.13 & 4.86 \times 10^4 & 4.75 \times 10^4 & 0 & 0 & 0 \\ -6.94 & 4.75 \times 10^4 & 7.85 \times 10^4 & 1.00 \times 10^4 & 0 & 0 \\ 0 & 0 & 1.00e \times 10^4 & 9.31 \times 10^4 & 7.94 \times 10^4 & 0 \\ 0 & 0 & 0 & 7.94 \times 10^4 & 9.75 \times 10^4 & 8.20 \times 10^2 \\ 0 & 0 & 0 & 0 & 8.20 \times 10^2 & 1.01 \times 10^5 \end{bmatrix}$$

and with the parasitic capacitances

$$\mathbf{K} = \begin{bmatrix} 6.69 \times 10^4 & 7.71 \times 10^2 & 7.35 \times 10^2 & 1.41 \times 10^1 & 1.34 \times 10^1 & 3.15 \times 10^{-1} \\ 7.71 \times 10^2 & 4.75 \times 10^4 & 4.62 \times 10^4 & 8.88 \times 10^2 & 8.45 \times 10^2 & 1.98 \times 10^1 \\ 7.35 \times 10^2 & 4.62 \times 10^4 & 7.69 \times 10^4 & 1.13 \times 10^4 & 1.22 \times 10^3 & 2.87 \times 10^1 \\ 1.41 \times 10^1 & 8.88 \times 10^2 & 1.13 \times 10^4 & 9.01 \times 10^4 & 7.62 \times 10^4 & 1.79 \times 10^3 \\ 1.34 \times 10^1 & 8.45 \times 10^2 & 1.22 \times 10^3 & 7.62 \times 10^4 & 9.39 \times 10^4 & 3.00 \times 10^3 \\ 3.15 \times 10^{-1} & 1.98 \times 10^1 & 2.87 \times 10^1 & 1.79 \times 10^3 & 3.00 \times 10^3 & 9.85 \times 10^4 \end{bmatrix}.$$

It can be seen that in the first case, the  $\mathbf{K}$  matrix has almost the tri-diagonal form (the values  $k_{13} = k_{31} = -6.94$  are caused by the capacitor  $C_2$ ) and hence it can be decomposed in the same way as was described in the previous section. But if the parasitic capacitances are considered, all matrix coefficients are non-zero and therefore the decomposition into blocks cannot be done without losing the accuracy (coefficients with small values have to be replaced by zeros).

Nevertheless, the circuit still has the feed-forward topology where there can be identified some blocks which are connected in one node with the current flowing between them and thus the decomposition into blocks should be still possible. This decomposition of the circuit schematic from Figure 5.20 is illustrated in Figure 5.21.

It consists of three blocks – tube amplifier stages, each containing one triode. The mutual interaction between the blocks is caused by connection currents  $i_{\text{conn1}}$  and  $i_{\text{conn2}}$  which are marked in Figure 5.21 with red color. If we consider the simulation of each tube amplifier stage separately, then these connection currents are actually additional current inputs of given tube amplifier stages. The nonlinear functions of the blocks can be written as

$$\mathbf{0} = \mathbf{G}_1 \mathbf{x}_1 + \mathbf{H}_1 \begin{pmatrix} v_{\text{in}} & v_{\text{ps}} & i_{\text{conn1}} \end{pmatrix}^T + \mathbf{K}_1 \mathbf{i}_1(\mathbf{v}_1) - \mathbf{v}_1 \quad (5.32)$$

$$\mathbf{0} = \mathbf{G}_2 \mathbf{x}_2 + \mathbf{H}_2 \begin{pmatrix} v_{\text{ps}} & i_{\text{conn1}} & i_{\text{conn2}} \end{pmatrix}^T + \mathbf{K}_2 \mathbf{i}_2(\mathbf{v}_2) - \mathbf{v}_2 \quad (5.33)$$

$$\mathbf{0} = \mathbf{G}_3 \mathbf{x}_3 + \mathbf{H}_3 \begin{pmatrix} v_{\text{ps2}} & i_{\text{conn2}} \end{pmatrix}^T + \mathbf{K}_3 \mathbf{i}_3(\mathbf{v}_3) - \mathbf{v}_3 \quad (5.34)$$

for unknown variables  $\mathbf{v}$  and matrices  $\mathbf{G}$ ,  $\mathbf{H}$  and  $\mathbf{K}$  which can be derived for each block. Then, if the blocks are connected together, the connection currents  $i_{\text{conn1}}$

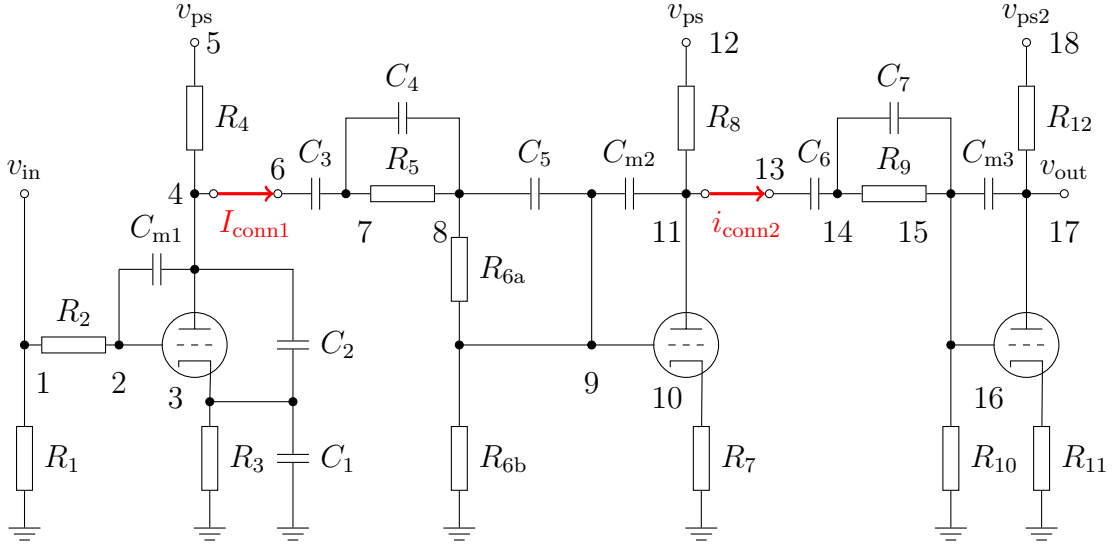


Figure 5.21: Circuit schematic of the Marshall JCM 800 guitar preamp with the decomposition into blocks.

and  $i_{\text{conn}2}$  are unknown but they cannot be found directly because there are three equations (5.32), (5.33), (5.34), each for one block, and only two unknown variables. Further conditions come from the fact that voltages of nodes  $V_4$ ,  $V_6$  resp.  $V_{11}$ ,  $V_{13}$  (see Figure 5.21), in which the blocks are connected, have to be equal. Therefore, we need to exploit the output equation of the DK-method. The voltage on the connection node of the first block can be expressed as

$$v_4 = \mathbf{D}_{11}\mathbf{x}_1 + \mathbf{E}_{11} \begin{pmatrix} v_{\text{in}} & v_{\text{ps}} & i_{\text{conn}1} \end{pmatrix}^T + \mathbf{F}_{11}\mathbf{i}_1(\mathbf{v}_1). \quad (5.35)$$

The second block has two connection nodes given by

$$v_6 = \mathbf{D}_{21}\mathbf{x}_2 + \mathbf{E}_{21} \begin{pmatrix} v_{\text{ps}} & i_{\text{conn}1} & i_{\text{conn}2} \end{pmatrix}^T + \mathbf{F}_{21}\mathbf{i}_2(\mathbf{v}_2), \quad (5.36)$$

$$v_{11} = \mathbf{D}_{22}\mathbf{x}_2 + \mathbf{E}_{22} \begin{pmatrix} v_{\text{ps}} & i_{\text{conn}1} & i_{\text{conn}2} \end{pmatrix}^T + \mathbf{F}_{22}\mathbf{i}_2(\mathbf{v}_2) \quad (5.37)$$

and the connection node of the third block is

$$v_{13} = \mathbf{D}_{31}\mathbf{x}_3 + \mathbf{E}_{31} \begin{pmatrix} v_{\text{ps}2} & i_{\text{conn}2} \end{pmatrix}^T + \mathbf{F}_{31}\mathbf{i}_3(\mathbf{v}_3). \quad (5.38)$$

Using the equality  $v_4 = v_6$  and  $v_{11} = v_{13}$ , one can obtain the final equations

$$0 = \mathbf{D}_{11}\mathbf{x}_1 + \mathbf{E}_{11} \begin{pmatrix} v_{\text{in}} & v_{\text{ps}} & i_{\text{conn}1} \end{pmatrix}^T + \mathbf{F}_{11}\mathbf{i}_1(\mathbf{v}_1) - \mathbf{D}_{21}\mathbf{x}_2 - \mathbf{E}_{21} \begin{pmatrix} v_{\text{ps}} & i_{\text{conn}1} & i_{\text{conn}2} \end{pmatrix}^T - \mathbf{F}_{21}\mathbf{i}_2(\mathbf{v}_2), \quad (5.39)$$

$$0 = \mathbf{D}_{22}\mathbf{x}_2 + \mathbf{E}_{22} \begin{pmatrix} v_{\text{ps}} & i_{\text{conn}1} & i_{\text{conn}2} \end{pmatrix}^T + \mathbf{F}_{22}\mathbf{i}_2(\mathbf{v}_2) - \mathbf{D}_{31}\mathbf{x}_3 - \mathbf{E}_{31} \begin{pmatrix} v_{\text{ps}2} & i_{\text{conn}2} \end{pmatrix}^T - \mathbf{F}_{31}\mathbf{i}_3(\mathbf{v}_3) \quad (5.40)$$

from which the connection currents  $i_{\text{conn1}}$  and  $i_{\text{conn2}}$  can be numerically computed. Each iteration however requires the numerical solution of (5.32), (5.33) and (5.34), which does not seem as an advantage so far. But on the other hand, the equation (5.32) can be rewritten into

$$\begin{aligned}
\mathbf{0} &= \mathbf{G}_1 \mathbf{x}_1 + \mathbf{H}_1 \begin{pmatrix} v_{\text{in}} & v_{\text{ps}} & i_{\text{conn1}} \end{pmatrix}^T + \mathbf{K}_1 \mathbf{i}_1(\mathbf{v}_1) - \mathbf{v}_1 = \\
&= \mathbf{G}_1 \mathbf{x}_1 + \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \end{bmatrix} \begin{bmatrix} v_{\text{in}} & v_{\text{ps}} & i_{\text{conn1}} \end{bmatrix}^T + \mathbf{K}_1 \mathbf{i}_1(\mathbf{v}_1) - \mathbf{v}_1 = \\
&= \mathbf{G}_1 \mathbf{x}_1 + \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{bmatrix} \begin{bmatrix} v_{\text{in}} & v_{\text{ps}} \end{bmatrix}^T + \begin{bmatrix} h_{13} \\ h_{23} \end{bmatrix} i_{\text{conn1}} + \mathbf{K}_1 \mathbf{i}_1(\mathbf{v}_1) - \mathbf{v}_1 = \quad (5.41) \\
&= \mathbf{p}_1 + \begin{bmatrix} h_{13} \\ h_{23} \end{bmatrix} i_{\text{conn1}} + \mathbf{K}_1 \mathbf{i}_1(\mathbf{v}_1) - \mathbf{v}_1 = \\
&= \overline{\mathbf{p}}_1 + \mathbf{K}_1 \mathbf{i}_1(\mathbf{v}_1) - \mathbf{v}_1
\end{aligned}$$

where the vector  $\overline{\mathbf{p}}_1$  contains two elements and therefore the equation (5.41) can be easily approximated as two-dimensional function  $\mathbf{i}_{1\text{app}}(\overline{\mathbf{p}}_1)$  where  $\overline{\mathbf{p}}_1 = \mathbf{p}_1 + \begin{bmatrix} h_{13} \\ h_{23} \end{bmatrix} i_{\text{conn1}}$ . The equations (5.33) and (5.34) can be approximated in the similar way, both requiring the 2D approximation. These approximating functions can be substituted into (5.39) and (5.40) to get equations

$$\begin{aligned}
0 &= \mathbf{D}_{11} \mathbf{x}_1 + \mathbf{E}_{11} \begin{pmatrix} v_{\text{in}} & v_{\text{ps}} & i_{\text{conn1}} \end{pmatrix}^T + \mathbf{F}_{11} \mathbf{i}_{1\text{app}} \left( \mathbf{p}_1 + \begin{bmatrix} h_{13} \\ h_{23} \end{bmatrix} i_{\text{conn1}} \right) - \\
&\quad - \mathbf{D}_{21} \mathbf{x}_2 - \mathbf{E}_{21} \begin{pmatrix} v_{\text{ps}} & i_{\text{conn1}} & i_{\text{conn2}} \end{pmatrix}^T - \\
&\quad - \mathbf{F}_{21} \mathbf{i}_{2\text{app}} \left( \mathbf{p}_2 + \begin{bmatrix} h_{12} & h_{13} \\ h_{22} & h_{23} \end{bmatrix} \begin{bmatrix} i_{\text{conn1}} \\ i_{\text{conn2}} \end{bmatrix} \right), \quad (5.42)
\end{aligned}$$

$$\begin{aligned}
0 &= \mathbf{D}_{22} \mathbf{x}_2 + \mathbf{E}_{22} \begin{pmatrix} v_{\text{ps}} & i_{\text{conn1}} & i_{\text{conn2}} \end{pmatrix}^T + \\
&\quad + \mathbf{F}_{22} \mathbf{i}_{2\text{app}} \left( \mathbf{p}_2 + \begin{bmatrix} h_{12} & h_{13} \\ h_{22} & h_{23} \end{bmatrix} \begin{bmatrix} i_{\text{conn1}} \\ i_{\text{conn2}} \end{bmatrix} \right) - \mathbf{D}_{31} \mathbf{x}_3 - \\
&\quad - \mathbf{E}_{31} \begin{pmatrix} v_{\text{ps2}} & i_{\text{conn2}} \end{pmatrix}^T - \mathbf{F}_{31} \mathbf{i}_{3\text{app}} \left( \mathbf{p}_3 + \begin{bmatrix} h_{13} \\ h_{23} \end{bmatrix} i_{\text{conn2}} \right). \quad (5.43)
\end{aligned}$$

The equations now only requires the numerical solving in real-time for two unknowns –  $i_{\text{conn1}}$ ,  $i_{\text{conn2}}$  instead of the numerical solving of original six unknown variables, which is a big improvement.

As it can be seen, the determination of nonlinear functions which are able to connect separate blocks is quite complicated and thus obtaining of connection functions in more straightforward way would be a great benefit. We will start with the analysis of the whole schematic instead of the analysis of separate blocks and further, we can define a nonlinear connection component with the connection current flowing through but zero voltage across the component terminals. The incorporation of this connection component however requires renumbering of the original circuit nodes (compare circuit schematics in Figure 5.20 and 5.21). The analysis of the whole circuit can be then done in the same way using incidence matrices as the analysis of the separate parts or any other circuit without these connection components. The only change is in the matrix  $\mathbf{N}_n$  which defines positions of nonlinear components and in which the connection components have to be added – each connection component as one new row in the matrix with 1 and  $-1$  for the nodes where the component is connected and 0 for the other nodes. The final incidence matrices are given in the appendix E and DK-method matrices can be computed in standard way. As a result, the  $\mathbf{K}$  matrix can be obtained in the form <sup>2</sup>

$$\mathbf{K} = \begin{bmatrix} k_{11} & k_{12} & 0 & 0 & 0 & 0 & k_{17} & 0 \\ k_{21} & k_{22} & 0 & 0 & 0 & 0 & k_{27} & 0 \\ 0 & 0 & k_{33} & k_{34} & 0 & 0 & k_{37} & k_{38} \\ 0 & 0 & k_{43} & k_{44} & 0 & 0 & k_{47} & k_{48} \\ 0 & 0 & 0 & 0 & k_{55} & k_{56} & 0 & k_{58} \\ 0 & 0 & 0 & 0 & k_{65} & k_{66} & 0 & k_{68} \\ k_{71} & k_{72} & k_{73} & k_{74} & 0 & 0 & k_{77} & k_{78} \\ 0 & 0 & k_{83} & k_{84} & k_{85} & k_{86} & k_{87} & k_{88} \end{bmatrix}.$$

One can see that tube amplifier stages are clearly separated. The first block is represented by the grid and plate current weighed by coefficients  $k_{11}$ ,  $k_{12}$ ,  $k_{21}$ ,  $k_{22}$  and the interaction with the second block is via the current  $i_{\text{conn1}}$  and coefficients  $k_{17}$ ,  $k_{27}$ . The second block consists of coefficients  $k_{33}$ ,  $k_{34}$ ,  $k_{43}$ ,  $k_{44}$  and weighed connection currents  $i_{\text{conn1}}$ ,  $i_{\text{conn2}}$  by coefficients  $k_{37}$ ,  $k_{38}$ ,  $k_{47}$ ,  $k_{48}$  and the third block consists of coefficients  $k_{55}$ ,  $k_{56}$ ,  $k_{65}$ ,  $k_{66}$  and weighed connection current  $i_{\text{conn2}}$  by coefficients  $k_{58}$ ,  $k_{68}$ . The 7-th and 8-th row of the matrix represent connection functions.

Until now, connection components were considered as standard circuit components but as it was stated above, the voltage across the component must be zero. Therefore, the standard DK-method nonlinearity given e.g. by the equation (3.104) have to be transformed into

$$\mathbf{0} = \mathbf{p} + \mathbf{K}\mathbf{i} - \mathbf{v}\mathbf{Z}_n \quad (5.44)$$

---

<sup>2</sup>the  $\mathbf{K}$  matrix with real values is in the appendix E.



where

$$\mathbf{Z}_n = \text{diag} \left( \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \end{bmatrix} \right) \quad (5.45)$$

where zeros are at positions of rows with connection components in the  $\mathbf{N}_n$  matrix and vectors  $\mathbf{v}$  and  $\mathbf{i}$  are given according to

$$\mathbf{v} = \begin{bmatrix} v_{gk1} & v_{pk1} & v_{gk2} & v_{pk2} & v_{gk3} & v_{pk3} & i_{\text{conn1}} & i_{\text{conn2}} \end{bmatrix}^T,$$

$$\mathbf{i} = \begin{bmatrix} i_g(v_1, v_2) & i_p(v_1, v_2) & i_g(v_3, v_4) & i_p(v_3, v_4) & i_g(v_5, v_6) & i_p(v_5, v_6) & v_7 & v_8 \end{bmatrix}^T.$$

The equation (5.44) can be solved directly but this does not make much sense because the nonlinearity is more complicated than the original (there are now two extra unknown variables). However, the nonlinearity can be easily decomposed into

$$\mathbf{0} = \begin{bmatrix} \bar{p}_1 \\ \bar{p}_2 \end{bmatrix} + \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix} \begin{bmatrix} i_g(v_1, v_2) \\ i_p(v_1, v_2) \end{bmatrix} - \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}, \quad (5.46)$$

$$\mathbf{0} = \begin{bmatrix} \bar{p}_3 \\ \bar{p}_4 \end{bmatrix} + \begin{bmatrix} k_{33} & k_{34} \\ k_{43} & k_{44} \end{bmatrix} \begin{bmatrix} i_g(v_3, v_4) \\ i_p(v_3, v_4) \end{bmatrix} - \begin{bmatrix} v_3 \\ v_4 \end{bmatrix}, \quad (5.47)$$

$$\mathbf{0} = \begin{bmatrix} \bar{p}_5 \\ \bar{p}_6 \end{bmatrix} + \begin{bmatrix} k_{55} & k_{56} \\ k_{65} & k_{66} \end{bmatrix} \begin{bmatrix} i_g(v_5, v_6) \\ i_p(v_5, v_6) \end{bmatrix} - \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \quad (5.48)$$

for arbitrary inputs  $\bar{p}_1, \bar{p}_2, \bar{p}_3, \bar{p}_4, \bar{p}_5, \bar{p}_6$  and nonlinear triode currents  $i_{g1}, i_{p1}, i_{g2}, i_{p2}, i_{g3}$  and  $i_{p3}$  can be approximated for these inputs. Finally, the nonlinear connection function is given by

$$\begin{bmatrix} k_{71} & 0 \\ k_{72} & 0 \\ k_{73} & k_{83} \\ k_{74} & k_{84} \\ 0 & k_{85} \\ 0 & k_{86} \\ k_{77} & k_{87} \\ k_{78} & k_{88} \end{bmatrix}^T \begin{bmatrix} i_{g1}(p_1 + k_{17}i_{\text{conn1}}, p_2 + k_{27}i_{\text{conn1}}) \\ i_{p1}(p_1 + k_{17}i_{\text{conn1}}, p_2 + k_{27}i_{\text{conn1}}) \\ i_{g2}(p_3 + k_{37}i_{\text{conn1}} + k_{38}i_{\text{conn2}}, p_4 + k_{47}i_{\text{conn1}} + k_{48}i_{\text{conn2}}) \\ i_{p2}(p_3 + k_{37}i_{\text{conn1}} + k_{38}i_{\text{conn1}}, p_4 + k_{47}i_{\text{conn1}} + k_{48}i_{\text{conn2}}) \\ i_{g3}(p_5 + k_{58}i_{\text{conn2}}, p_6 + k_{68}i_{\text{conn2}}) \\ i_{p3}(p_5 + k_{58}i_{\text{conn2}}, p_6 + k_{68}i_{\text{conn2}}) \\ i_{\text{conn1}} \\ i_{\text{conn2}} \end{bmatrix} = - \begin{bmatrix} p_7 \\ p_8 \end{bmatrix} \quad (5.49)$$

with two unknown variables  $i_{\text{conn1}}$  and  $i_{\text{conn2}}$  to be solved numerically.

The validity of the block decomposition was tested by the comparison of the output signals of the simulation without the block decomposition and the simulation with this block decomposition. Both output signals as well as their difference are shown in Figure 5.22. The input signal was a sine wave with a frequency of 500 Hz and an amplitude of 0.1 V. The approach was also tested using other input signals and all showed similar results. The algorithm was also tested in real-time conditions.

The computational cost of the Marshall JCM 800 simulation with the connection currents implemented using the C language was between 10 and 15 % of CPU on the Intel 2.6 GHz processor.

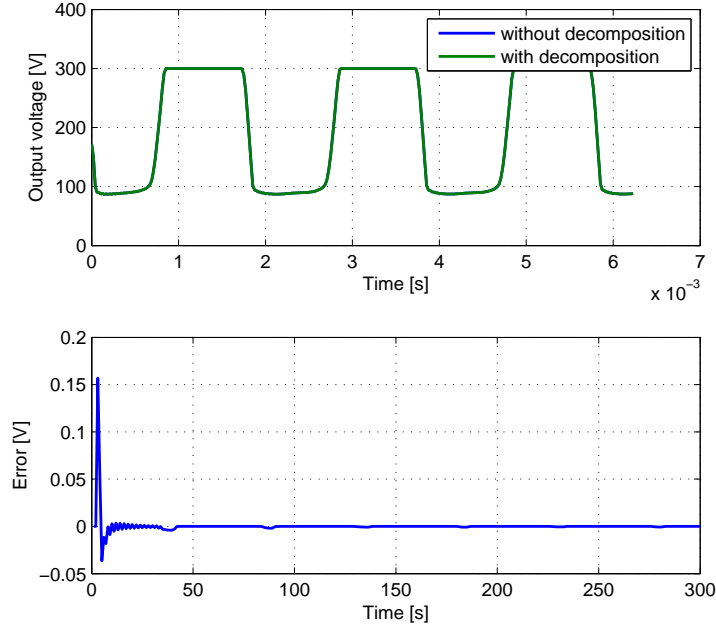


Figure 5.22: Output signals (top) and their difference (bottom) for the simulation of Marshall preamp with and without the decomposition

## 5.5 Simulation of Circuit with Global Feedback

There are many audio effects or guitar amplifiers which do not have the strict feed-forward topology and they contain a global feedback. Examples of these effects can be effect such as delay, flanger, phaser, feedback compressor. The frequency dependent feedback can also be found in tube push-pull amplifiers. The global feedback causes that the whole circuit contains the delay free loop and therefore the whole circuit must be solved without any decomposition. This problem can be circumvented if a one sample delay block is connected in the feedback. This can be done without losing the accuracy for audio effects which use the delay line, e.g. delay or flanger, because the delay has already been introduced there. Firstly, if the linear circuit without the delay and the instantaneous global feedback is considered, the transfer function can be expressed as

$$H(z) = \frac{H_{ff}(z)}{1 + H_{ff}(z)H_{fb}(z)} \quad (5.50)$$

and with the delayed feedback

$$H(z) = \frac{H_{\text{ff}}(z)}{1 + z^{-1}H_{\text{ff}}(z)H_{\text{fb}}(z)} \quad (5.51)$$

which means that the frequency response of the audio effect is affected. The impact of the delayed global feedback can be found directly by comparing both transfer functions (5.50) and (5.51). Similar behavior can be expected with nonlinear systems as well.

To find out the impact of the delayed global feedback on the nonlinear system, the guitar tube power amplifier was chosen as a case study. The typical circuit schematic of the tube power amplifier is shown in Figure 5.23 and values of the components are given in Table 5.15 [87]. This circuit is without any simplifications and consists of the triode phase splitter and the push-pull amplifier, which has already been discussed earlier concerning different types of the output transformer. The output transformer here was supplemented with parasitic capacitances  $C_7$ ,  $C_8$ ,  $C_9$  and leakage inductances  $L_1$  and  $L_2$ . The power amplifier was also supplemented with the model of the loudspeaker. The loudspeaker is modeled with the electric impedance model with parameters derived from the measurement of the Celestion Vintage 30 loudspeaker using the added mass method and Thiele/Small parameters [96]. Miller parasitic capacitances, marked with blue color, were added too. The variable global feedback between  $V_{\text{out}}$  and  $V_{\text{fdb}}$  is marked with red color and it is adjustable via the resistor  $R_7$ .

Table 5.15: Component values of the tube power amplifier.

Component	Value	Component	Value	Component	Value
$R_1, R_{17}, R_{18}$	10 $\Omega$	$R_2, R_4$	1 M $\Omega$	$R_3$	470 $\Omega$
$R_5$	10 k $\Omega$	$R_6$	4.7 k $\Omega$	$R_7, R_{10}$	100 k $\Omega$
$R_8$	22 k $\Omega$	$R_9$	82 k $\Omega$	$R_{11}, R_{12}$	220 k $\Omega$
$R_{13}, R_{14}$	5 k $\Omega$	$R_{15}, R_{16}$	1 k $\Omega$	$R_{19}$	6.8 $\Omega$
$R_{20}$	45 $\Omega$	$C_1, C_4, C_5$	22 nF	$C_2, C_3$	100 nF
$C_6$	1 pF	$C_7, C_8, C_9$	1 nF	$C_{10}$	193 $\mu$ F
$L_1, L_2$	10 mH	$L_3$	0.6 mH	$L_4$	23 $\mu$ F
$C_{m1}, C_{m2}, C_{m3}, C_{m4}$	1.6 pF	$V_{\text{ps}}$	430 V	$V_{\text{ps2}}$	450 V
$N_1$	1000	$N_2$	1000	$N_3$	50

The whole power amplifier circuit was simulated using the DK-method for four different topologies:

- The input feedback voltage  $v_{\text{fdb}}$  is the additional input of the circuit and it is obtained from the output voltage  $v_{\text{out}}$  delayed by one sample. The Miller

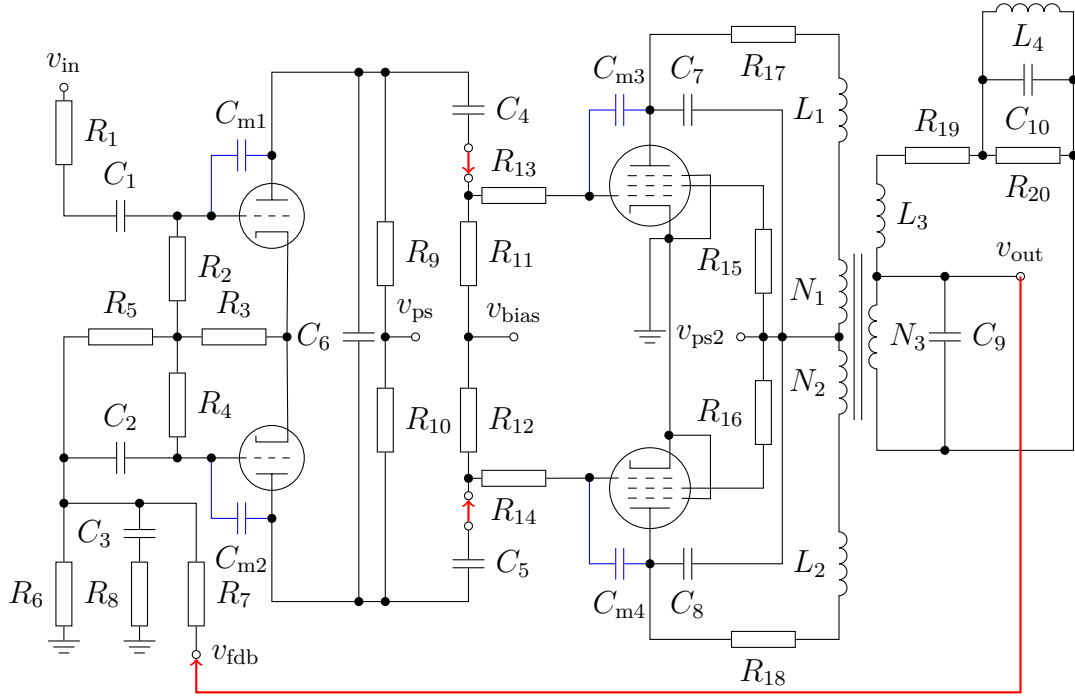


Figure 5.23: Circuit schematic of the tube power amplifier.

capacitances are unconnected. The coefficients of the  $\mathbf{K}$  matrix are separated for the phase splitter part and the push-pull amplifier part. The  $\mathbf{K}$  matrix has following form

$$\mathbf{K} = \begin{bmatrix} k_{11} & k_{12} & k_{13} & k_{14} & k_{15} & k_{16} & 0 & 0 & 0 & 0 \\ k_{21} & k_{22} & k_{23} & k_{24} & k_{25} & k_{26} & 0 & 0 & 0 & 0 \\ k_{31} & k_{32} & k_{33} & k_{34} & k_{35} & k_{36} & 0 & 0 & 0 & 0 \\ k_{41} & k_{42} & k_{43} & k_{44} & k_{45} & k_{46} & 0 & 0 & 0 & 0 \\ k_{51} & k_{52} & k_{53} & k_{54} & k_{55} & k_{56} & 0 & 0 & 0 & 0 \\ k_{61} & k_{62} & k_{63} & k_{64} & k_{65} & k_{66} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & k_{77} & k_{78} & k_{79} & k_{710} \\ 0 & 0 & 0 & 0 & 0 & 0 & k_{87} & k_{88} & k_{89} & k_{810} \\ 0 & 0 & 0 & 0 & 0 & 0 & k_{97} & k_{98} & k_{99} & k_{910} \\ 0 & 0 & 0 & 0 & 0 & 0 & k_{107} & k_{108} & k_{109} & k_{1010} \end{bmatrix}.$$

This enables the decomposition into two separate parts which can be approximated to work efficiently in real-time using the linear interpolation.

- The input feedback voltage  $v_{fdb}$  is the additional input of the circuit and it is obtained from the output voltage  $v_{out}$  delayed by one sample but Miller capacitances are connected. Because of the feedback coupling caused by pentode Miller capacitances  $C_{m3}$  and  $C_{m3}$ , the  $\mathbf{K}$  matrix coefficients are not separated for the phase splitter and the push-pull part and thus the circuit must be

simulated as one system or connection components designed in the previous section can be connected between  $C_4$ ,  $R_{13}$  and  $C_5$ ,  $R_{14}$  (marked with red arrows in Figure 5.23). Then it requires only the numerical solving for two unknown variables after the phase splitter and the push-pull part are approximated.

- The input feedback voltage  $v_{\text{fdb}}$  node is directly connected with the output voltage node  $v_{\text{out}}$ . Miller capacitances are unconnected. Although there is no feedback coupling caused by pentode Miller capacitances, the phase splitter and the push-pull part are still coupled via the global feedback and the decomposition is only possible using the connection components. It requires only one connection component representing the feedback current and therefore, after the approximation, only numerical solving for one unknown variable is needed.
- The input feedback voltage  $v_{\text{fdb}}$  node is directly connected with the output voltage node  $v_{\text{out}}$  and Miller capacitances are connected. The decomposition is only possible using three connection components – the feedback current and feed-forward currents between the phase splitter and the push-pull part. After the approximation of these parts, it requires numerical solving for three unknown variables in real-time.

All four topologies were simulated without the block decomposition and were compared with the four topologies decomposed using three connection components. Foremost, the impact of the delayed global feedback and Miller capacitances to the transfer function was investigated. Therefore, the frequency response magnitude of the first harmonic component is shown in Figures 5.24 and 5.25. Figure 5.24 shows the frequency response magnitude of the first harmonic component for the circuit with the parameter "Presence" set to zero (the resistance of the potentiometer  $R_7$  is set to full 22 k $\Omega$ ) and Figure 5.25 shows the frequency response for the parameter "Presence" set to one (the resistance of the potentiometer  $R_7$  is set to zero).

It can be seen that the delayed global feedback affects the transfer function only for high frequencies but these are still in the audible area. Therefore, the accurate simulation should contain the not-delayed feedback which, although it requires the numerical algorithm to solve, is capable to work in real-time because it only has one unknown variable – the feedback current. The impact of Miller capacitances is rather small and moreover it is much more computational consuming because it requires three unknown variables to be solved numerically in real-time. Omitting the Miller capacitances can be further justified by the fact that there is a loudspeaker cabinet connected to the power amplifier. These loudspeaker cabinets typically have the frequency bandwidth up to 7 or 8 kHz as it can be seen from Figure 5.26 where measured frequency responses of two loudspeaker cabinets are shown. Therefore, the small variations between the simplified and full model can be masked.

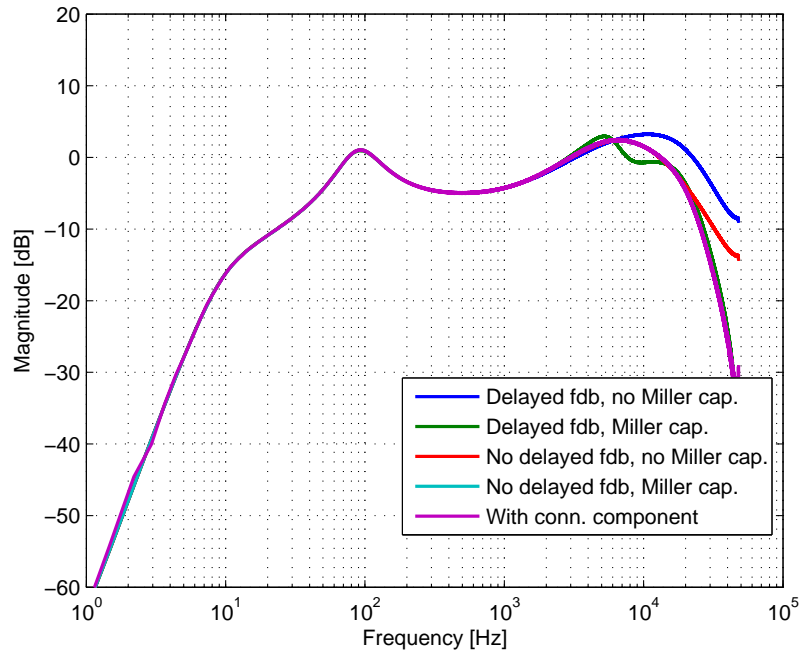


Figure 5.24: Frequency response magnitude of the first harmonic component for different topologies of the circuit for parameter "Presence" set to zero.

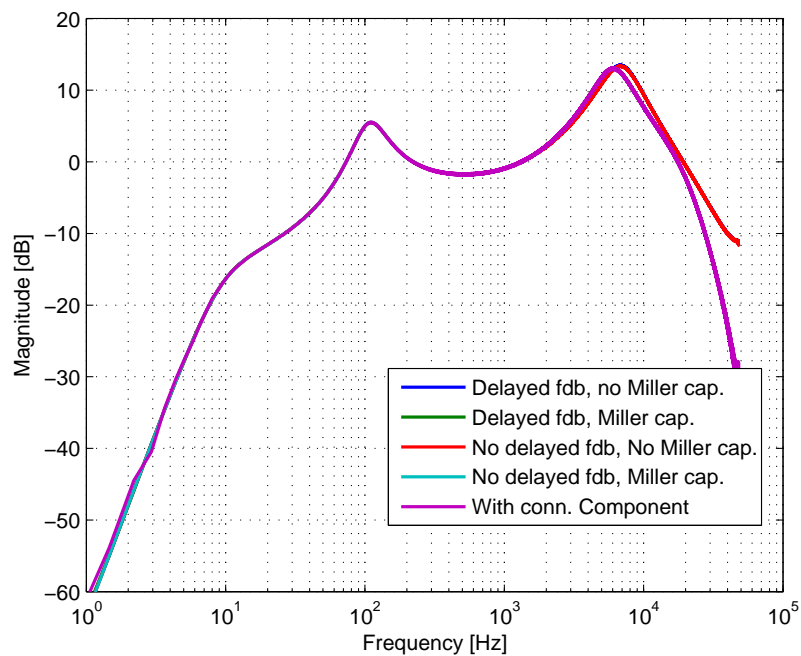


Figure 5.25: Frequency response magnitude of the first harmonic component for different topologies of the circuit for parameter "Presence" set to one.

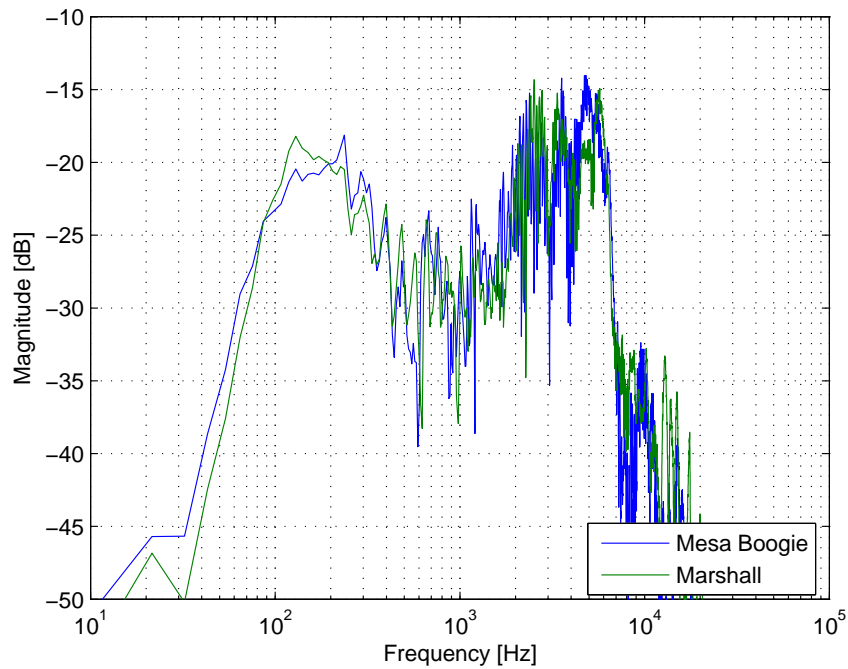


Figure 5.26: Frequency response magnitude of measured loudspeaker cabinets – the Mesa Boogie Rectifier and the Marshall JCM 800.

## 5.6 Summary

This chapter dealt with the real-time simulation of more complex audio effects and guitar amplifiers. Three different methods were designed, each with different computational complexity and different accuracy of the simulation. All proposed methods make use of the approximation of simpler blocks and these methods are focused foremost on the accurate connection of these blocks.

The first method was the modified block-wise method. It is an extension of simple decomposition into blocks. The main idea is connecting two neighbouring blocks into couples where the first block of the pair is the simulated block and the other one builds only the nonlinear load for the first block. This method can be efficiently applied for feed-forward audio effect circuits and it is very suitable for the implementation in DSP systems with lower computational power. This method was also implemented in the first versions of the commercial guitar amplifier simulator AmpLion<sup>3</sup>.

<sup>3</sup><http://www.audiffex.com/EN/amplion.html>

The second method is designed to work with the DK-method. Similarly to the previous method, this method works with feed-forward audio effect circuits and it is based on the similar idea that the latter blocks have small influence to the former block if they are not connected directly. Further it was shown that some blocks of the circuit can be almost independent on DK-method input variables. This fact enables reducing the dimension of approximating functions and thus approximating functions can be faster and they can require a smaller amount of coefficients stored in memory.

The third method is designed for the simulation of complex systems which contain feedbacks or very tight coupling of nonlinear devices. This method supposes that elementary sub-circuits are approximated for inner variables but also contain input for the current flowing from or into another connected block. Consequently, this method numerically solves unknown currents flowing between connected blocks in real-time. This approach enables significant reduction of the order of nonlinear functions. As a result, complex systems can be numerically solved in real-time while the accuracy of the simulation remain the same as for the numerical solution of the whole system without the decomposition.



## 6 QUALITY OF SIMULATION OF AUDIO EFFECT CIRCUITS

The evaluation of the simulation quality is inseparable part of the process of designing the simulation algorithm. Quality of proposed algorithms will be discussed in this chapter – it will include comparison of simulated and measured output signals in the time and frequency domain, which is the first step when designing the simulation algorithm. The designed algorithm will subsequently undergo subjective testing which will consist of several listening tests. The listening test will be focused on:

- evaluation of quality of the guitar tube preamp simulation,
- comparison of the output transformer models for use in guitar tube power amplifiers,
- comparison of the approximation of the nonlinear equations and numerical solving the nonlinear equations,
- audible aliasing distortion.

The listening test will be designed using theory given in literature [97]. This theory is quite complex and therefore it will not be discussed in this thesis.

### 6.1 Simulation of the Guitar Tube Preamp Engl E530

The quality of the simulation will be tested on the guitar preamp Engl E530 simulation. In order to build the most precise model, the circuit schematic was obtained using the reverse engineering of the real hardware device. The circuit schematic is, however, not published in this work due to copyrights. The next step was measuring the tubes because the tube model is one of the most important parts of the model and it can significantly affect the overall quality of the simulation of the particular hardware device. Tubes from the preamp were measured using the measurement device designed by Ch. Dempwolf and O. Kröning [98, 99] which enables automated measurement of the tubes with the given step of tube terminal voltages. From this data, different tube models can be designed using e.g. least mean square fitting between the tube model and measured data. Measured data can be also approximated e.g. using the piece-wise cubic spline interpolation to implement the tube model directly from measured data, which offers the most accurate tube model. This model has the advantage of continuous derivatives which are needed by numerical algorithm.

The software model of the preamp was constructed using the DK-method and was supplemented with additional capacitors to model Miller capacitances. The model allows later decomposition into several blocks containing one tube each. But

firstly, the numerical solving of the whole circuit was used to find out the quality of the proposed model.

Two channels – the clean channel and the crunch channel – of the preamp were simulated and several input testing signals, namely the sine wave, the sweep signal as well as the real guitar riff, were used. Figures 6.1 and 6.2 show the comparison of measured and simulated signals at plate terminals of the simulated preamp clean channel for the input sine wave with the frequency of 1 kHz and amplitudes of 0.5 and 2 V. The sampling frequency was 44.1 kHz. The measured signals were obtained using the oscilloscope and the signal generator. Both figures show good match between the simulated and measured signal. Figure 6.3 shows the frequency response magnitude of the first harmonic content. The frequency response was measured using the swept sine technique [54] and the sound card M-Audio Fasttrack. Figures 6.4 and 6.5 show spectrograms of output signals (measured and simulated). The input signal was swept sine signal with the duration of 100 s and the maximal value of 0.01 V. The sampling frequency was 44.1 kHz, the simulation used  $8 \times$  oversampling. The output signal from Figure 6.4 contains some hum noise (50 Hz and higher harmonics up to 5-th) which is not presented in the simulation. The output signal from the simulation algorithm still contains aliasing distortion, as it can be seen from Figure 6.5.

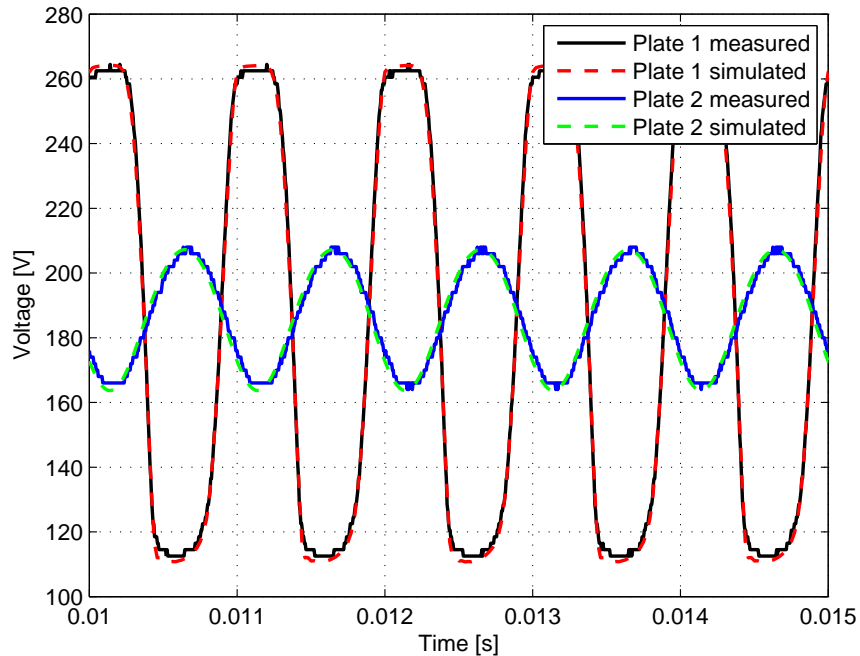


Figure 6.1: Comparison of measured and simulated voltage signals at plate terminals of the tubes for input sine wave with the amplitude 0.5 V.

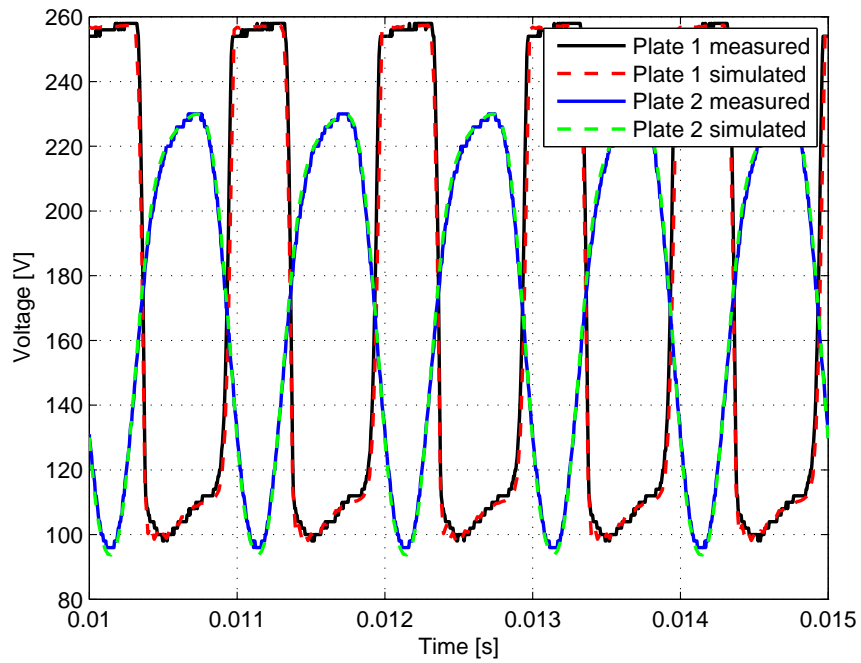


Figure 6.2: Comparison of measured and simulated voltage signals at plate terminals of the tubes for input sine wave with the amplitude 2 V.

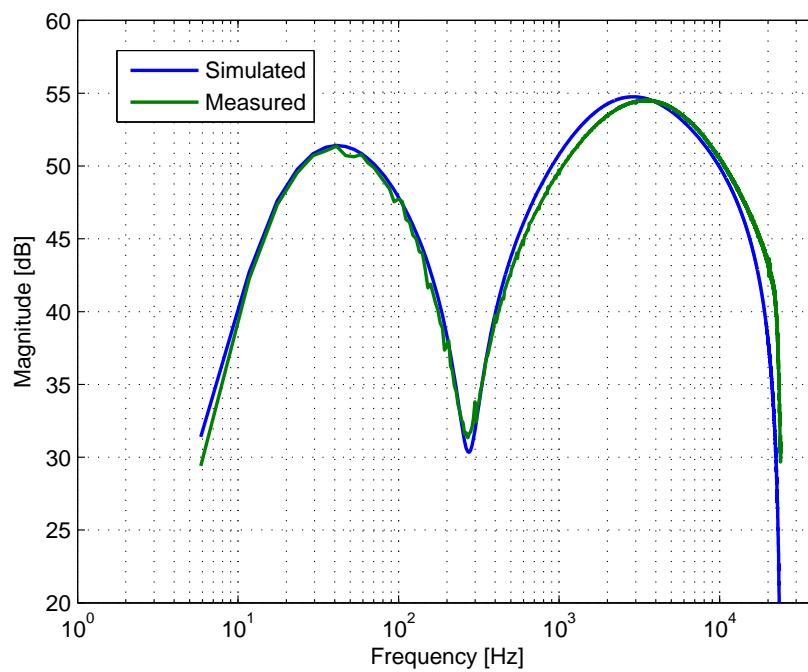


Figure 6.3: Comparison of measured and simulated frequency responses (for the first harmonic content).

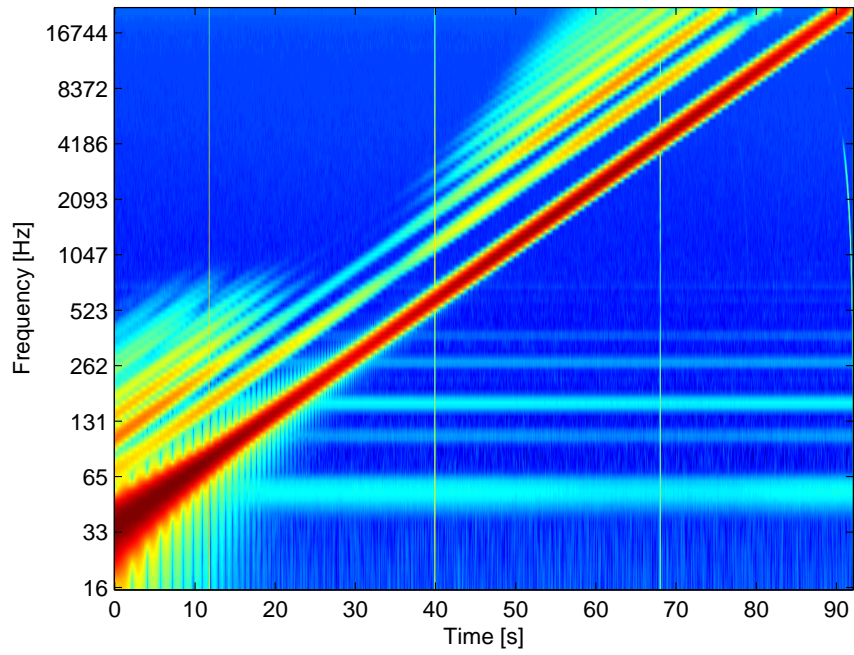


Figure 6.4: Spectrogram of the output signal - measured clean channel.

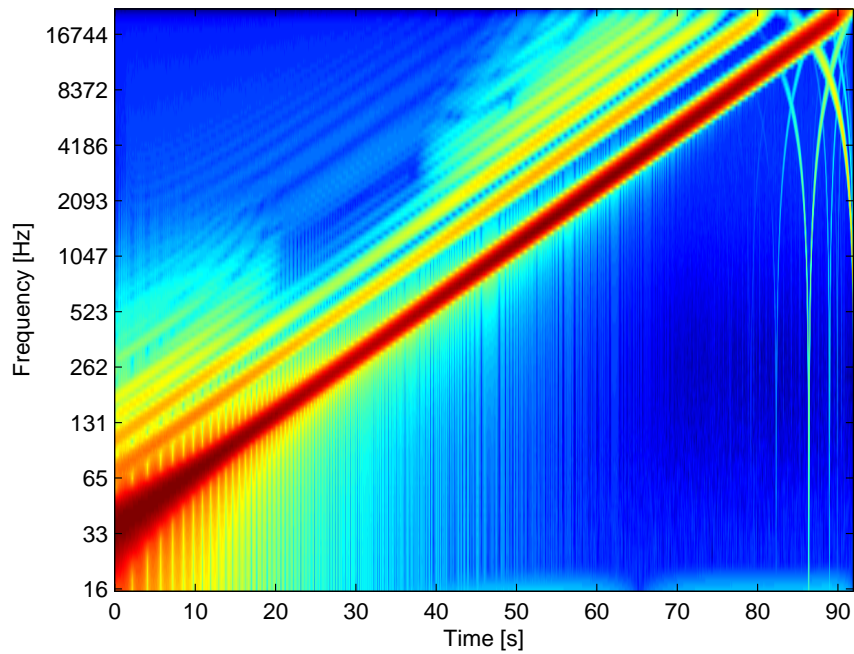


Figure 6.5: Spectrogram of the output signal - simulated clean channel.

The crunch channel was analyzed in similar way. The crunch channel consists of three triodes and the tone stack connected between the first and the second tube. The clean channel has only two triodes and therefore the crunch channel has more of nonlinear distortion. The comparison of the measured and simulated signal for the input sine wave signal at plate terminals of all three triodes is shown in Figures 6.6 and 6.6. Results show that the output signal from the simulation is very close to the measured one, however the difference at the third plate voltage is visible. This difference was caused by small shifting of the operating point of the third triode in the dynamic mode. Figure 6.8 shows the frequency response magnitude of the first harmonic content of the measured preamp and its simulation. Figures 6.9 and 6.10 show spectrograms of output signals (measured and simulated).  $16 \times$  oversampling was used to reduce the aliasing distortion in the simulation algorithm.

All figures showed that the proposed model offers the accurate simulation although output signals of the real device and the simulation algorithm are not identical. However, all circuit components have tolerance in their values and therefore values used in the model can be fitted to get better results. The tube model is not probably accurate enough as well. Therefore, the next step is a verification using the listening test.

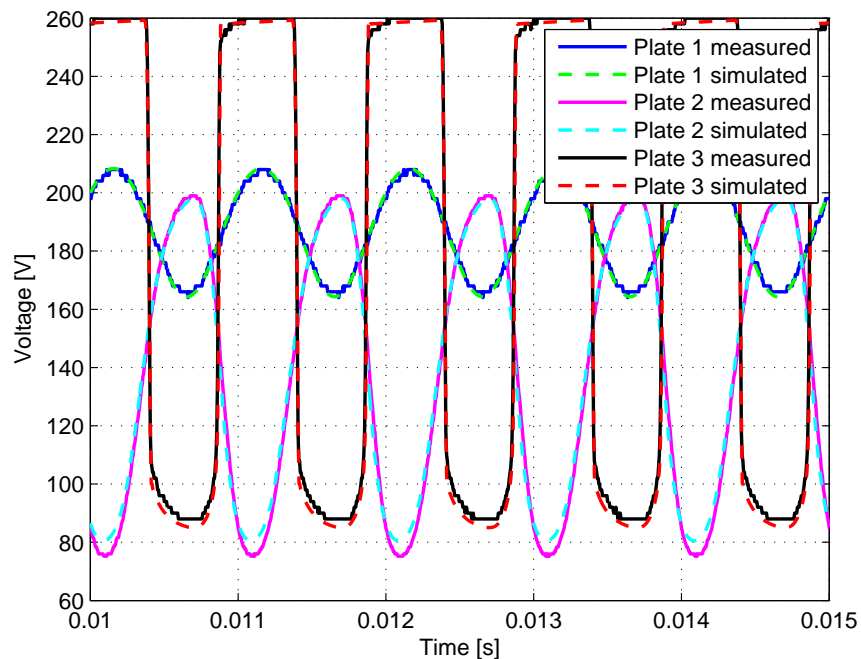


Figure 6.6: Comparison of measured and simulated voltage signals at plate terminals of the tubes for input sine wave with the amplitude 0.5 V.

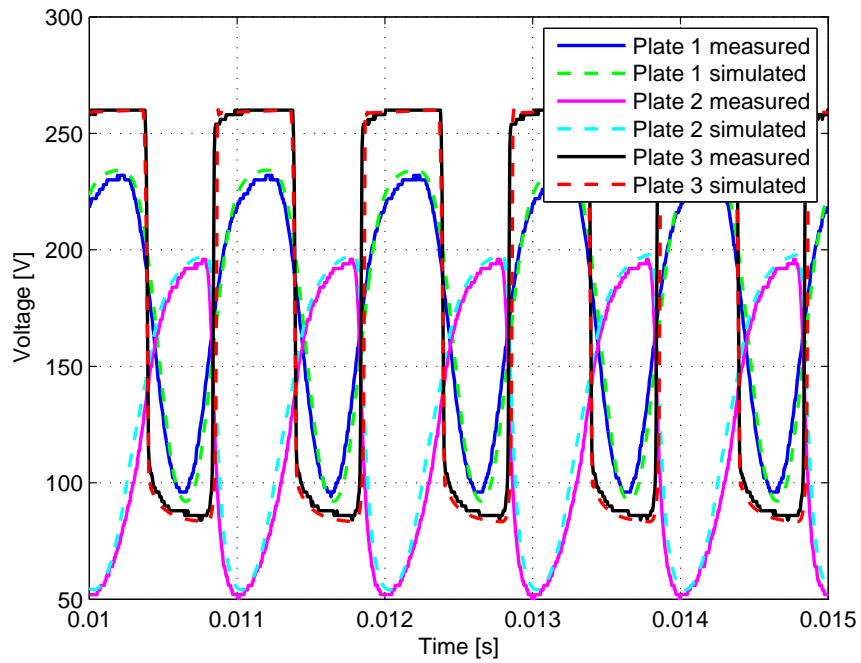


Figure 6.7: Comparison of measured and simulated voltage signals at plate terminals of the tubes for input sine wave with the amplitude 2 V.

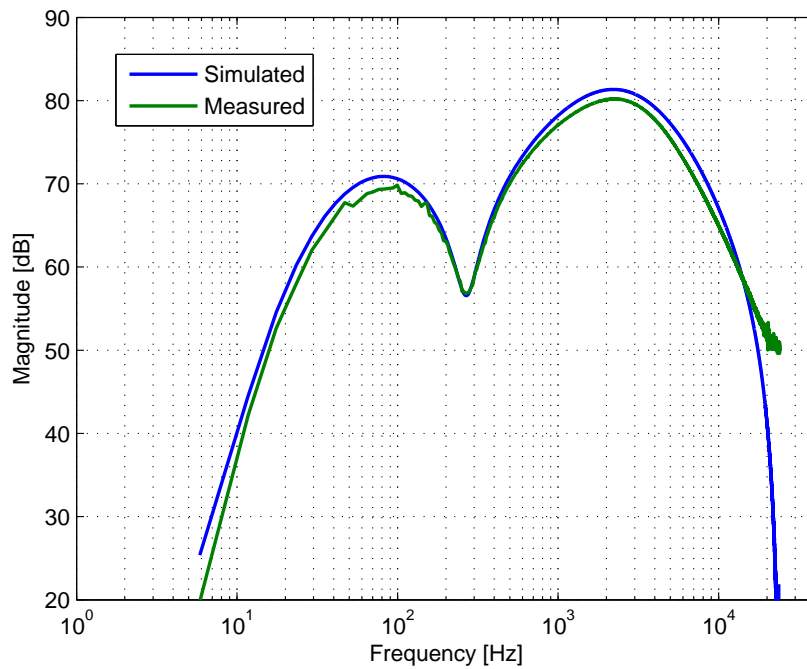


Figure 6.8: Comparison of measured and simulated frequency responses (for the first harmonic content).

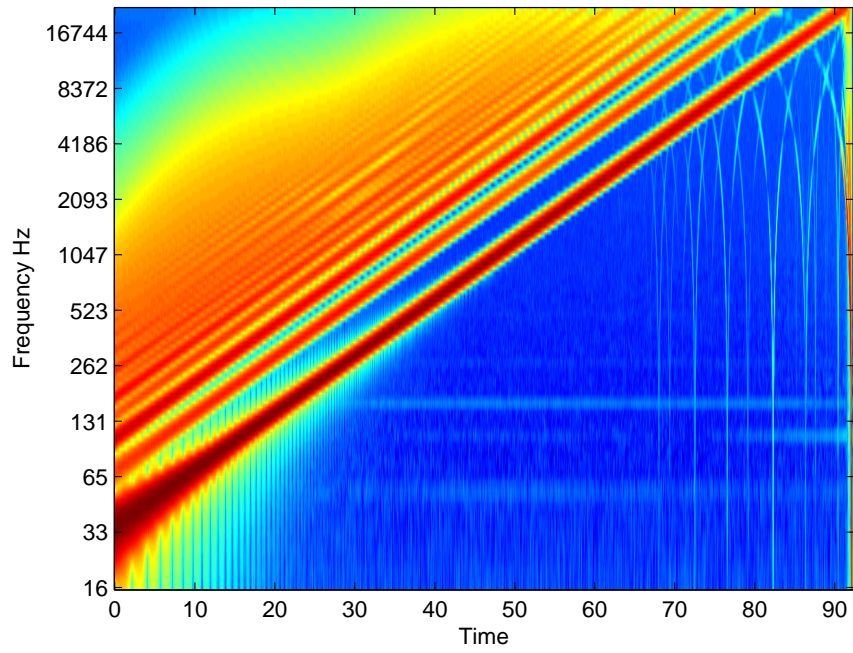


Figure 6.9: Spectrogram of the output signal - measured crunch channel.

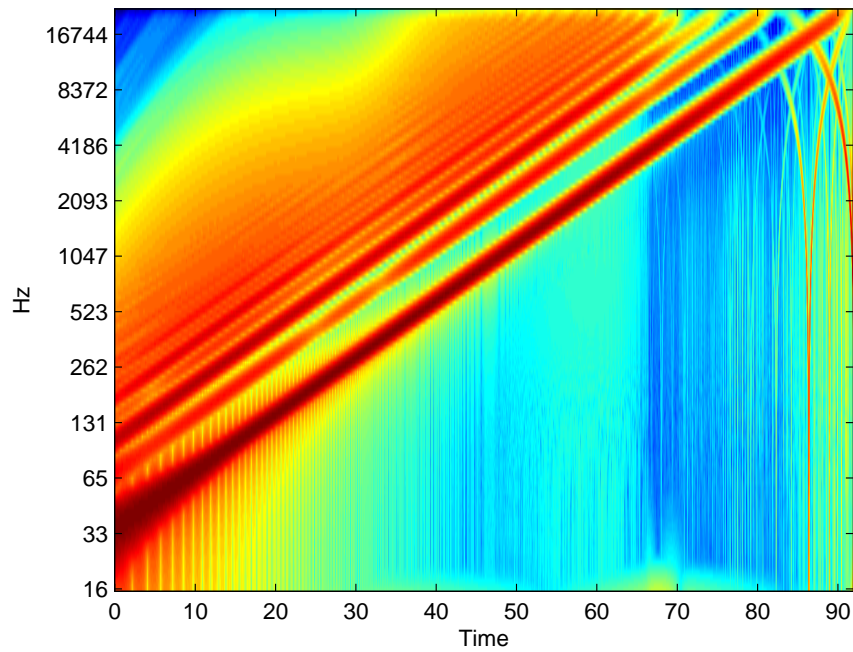


Figure 6.10: Spectrogram of the output signal - simulated crunch channel.

## 6.2 Subjective evaluation of guitar tube preamp simulation

When talking about the quality of the simulation of audio effects, the subjective evaluation of quality is more important than the objective evaluation. Unfortunately, the subjective quality evaluation can not be obtained as simply as the objective evaluation and the process of subjective evaluation of the quality requires listening tests with several people. Results of the listening test must be then processed statistically to find final results of the test. There are more types of listening tests, each suitable for different applications. We need the comparison of two sounds – one measured and one simulated and therefore ABX Double Blind Test, commonly used for the assessment of small impairments, e.g. the quality evaluation of audio codecs, was chosen [100, 101, 102]. The principle of the method is following. There are three sound samples with small impairments, samples A and B are reference samples (A is the measured signal and B is the simulated signal) and the sample X is randomly chosen sample A or B. Listeners are asked to recognize whether the sample X is the reference sample A or B. Listeners's answers are classified into two categories – correct and incorrect answers and they are tested statistically.

The listening test was divided into these stages:

1. **Generation of testing audio signals** – The ABX listen test requires audio signals with very small impairments otherwise it would be very easy to recognize the unknown sound sample X. Therefore, the input signal for the real device and the simulation algorithm must be exactly the same. It should be also tested with wide variety of real signals. Therefore, 10 different guitar riffs of several playing styles were recorded directly into the computer. The length of the riffs was around 20 s. The audio output port of the computer was connected to the measured guitar preamp. Firstly a testing sine wave signal was played and measured at the input and the output of the guitar preamp to find levels of the input and the output signal. Subsequently, all the guitar signals were played and output signals from the guitar preamp were again recorded into the computer. Dry guitar riffs were normalized to the same level which was at the input of the guitar preamp and then they were processed by the algorithm which simulates the guitar preamp. The  $8 \times$  oversampling was used to reduce aliasing distortion. Output signals from the simulation were normalized to the same level as recorded output signals of the real guitar preamp. Although very good match between the measured and the simulated signal was achieved, the signals were recognizable due to the noise generated by the guitar preamp itself, which was missing in the simulation algorithm.



Therefore, this noise was recorded with no input signal and it was added to the simulated signals. Finally, the triplets A,B,X were assembled and stored in multichannel wav sound files denoted as A1, A2, ..., A10 for the clean channel of the guitar preamp and B1, B2, ..., B10 for the crunch channel.

2. **Preparation of necessary equipment (hardware and software)** – Hardware used for the listening test consisted of the laptop, the external sound card M-Audio Fasttrack, and headphones AKG K-77. The laptop was equipped with software Cubase 5<sup>1</sup> which allows the playback of multichannel wav files. Because only one channel from the wav file can be played in a time, the VST plug-in effect which enables routing of the selected input channel to all output channels was implemented. This plug-in effect allows switching between signals A, B and X during the playback and thus, the judgment is more comfortable.
3. **Preparation of the process of the listening test** – Firstly, instructions for the listening test were written in Czech and English language. The English instructions can be found in Appendix G. Because the used software does not allow the automatic acquisition of listener's answers, an answer form was used. The answer form can be found in Appendix H. The next phase was the pretesting. This phase is important because potential shortcomings of the designed test can be revealed during this phase. This phase also can show time requirements. The judgment of all 20 files requires approximately 35 minutes.
4. **Performing of the listening test** – The listening test was performed with each listener individually. Each listener was familiarized with the software and hardware and the test's instructions. Subsequently, it was shown how to judge the files. Because this listening test is of the *forced choice* type, listeners were told that they have to choose the answer randomly if they are not sure about the answer. The listening test was a partially controlled experiment. Listeners worked independently and were able to adjust the tempo to their abilities but they had chance to ask the experiment leader if they were not sure.
5. **Processing of the obtained data** – Answers were corrected and the number of correct and incorrect answers was determined. The next phase is the statistical testing of results. Two statistical hypothesis were introduced:
  - $H_0$  – Differences between the measured and the simulated signal are not audible and listeners have to guess the answer. The probability of the correct answer is  $p_0 = 0.5$ .
  - $H_1$  – Differences between the measured and the simulated signal are audible. The probability of the correct answer  $p_1 > 0.5$ .

Because there are only two possible results of the experiment (correct or

---

<sup>1</sup><http://www.steinberg.net/de/products/cubase/>

incorrect identification), the probability distribution of the identifications is determined by the binomial distribution [102]

$$P(n, k, p) = \binom{n}{k} p^k (1 - p)^{n-k} \quad (6.1)$$

with probability  $p = 0.5$  and the number of experiments  $n$  and the number of hits  $k$ . It is possible to derive the number of correct identifications for the given level of significance  $\alpha = 0.05$  which are needed to reject the null hypothesis  $H_0$ .

### 6.2.1 Results

The listening test had 15 participants – guitarists, professional musicians, sound and audio engineers. Each listener had to identify 10 sound samples for each channel. Thus, the total number of independent experiments per channel was 150. Using the level of significance  $\alpha = 0.05$ , the total number of correct identifications needed to reject the null hypothesis is 85 and more (probability distribution  $> 0.95$ ). Table 6.1 shows results of the test – the number of correct identifications as well as the average difficulty to distinguish the simulated and the measured signal on the scale 1 – 5 where 5 is the most difficult.

Table 6.1: Results of subjective evaluation of preamp simulation.

Channel	Correct identifications	Difficulty to judge
Clean	83	4.41
Crunch	90	4.60

Results showed very good quality of the simulation. The number of correct identifications of the clean channel of the preamp was not sufficient to reject the null hypothesis and therefore we can say that measured and simulated signals are not distinguishable. The number of correct identifications of the crunch channel was sufficient to reject the null hypothesis. It means that signals are distinguishable but it requires big effort due to the high value of difficulty. These results are related to the whole group of listeners. From the individual listener point of view, results depend on abilities of each listener. Most of them were guessing and sometimes were able to distinguish the samples. Only two listeners were able to differ the samples correctly and name or describe the sound artefacts which were different for the samples. It was the different bass frequency response of the clean channel and the different compression of the output signal and mistuned higher harmonics of

the crunch preamp. The bass frequency response could be fitted by changing the components values. The different compression can be caused by use of the simplified power supply circuit. The model should be supplemented with a diode rectifier model instead of the constant voltage used in the model. The problem of the mistuned higher harmonics can come from the presence of aliasing distortion. Although  $8 \times$  oversampling was used, it does not have to be enough for some people.

### 6.3 Subjective Comparison of Interpolation Techniques

Chapters 4.3 and 4.4 were focused on the objective comparison of several interpolation techniques. All techniques provides very good function approximation if sufficient amount of data is used. The total amount of data can be significantly reduced using the nonuniform interpolation and algorithms 1 and 2. The amount of data depends on the given approximation error. But the permitted value of this error, for which is the result not audible, is unknown. Therefore, the listening test was designed to investigate this error. The VST plug-in effect with the simulation of the tube amplifier was implemented. This plug-in effect uses the DK-method and the nonlinear implicit equation is substituted with the approximated nonlinear plate current of the triode. The dimension of the approximation is 2. It is possible to choose which type of interpolation described in chapters 4.3 and 4.4 is used. It is also possible to change the amount of data used for the interpolation. Original data were reduced using the algorithm 2 and the reduced data from each iteration were stored under the given iteration index. The higher index means the higher approximation error but lower amount of data to be stored. The listener can change the value of the index using the slider control in the plug-in Graphical User Interface (GUI). The listeners were asked to increase the index until they were able to hear difference between the reduced and full set of data for each type of the interpolation. After that, they had to write the values into the answer form (see Appendix H). Results are available in Table 6.2.

The audible interpolation error is around  $7 \times 10^{-4}$  mA, only the Hermite cubic interpolation permits higher value. The cubic spline type 1 has the lower data reduction in this case (interestingly it is in contrast with the 1D approximation in Figure 4.4 and Table 4.3). All other interpolation types provides very similar results and therefore one can choose the interpolation type according to the computational cost. The best interpolation types are: the Hermite cubic interpolation if the amount of data should be minimal or the linear interpolation if the computational cost should be minimal.

Table 6.2: Results of the comparison of interpolation techniques using the listening test.

Interpolation	Average audible error	Number of points	number of coefficients
Spline 1	$7.05 \times 10^{-4}$	1872	29952
Spline 2	$7.26 \times 10^{-4}$	720	1440
Hermite	$13.18 \times 10^{-4}$	690	690
Newton	$9.09 \times 10^{-4}$	539	539
Linear	$6.28 \times 10^{-4}$	714	714

## 6.4 Subjective Comparison of Output Transformer Model

Three types of the output transformer were mentioned in chapter 3.1. It was shown that the nonlinear output transformer model affects low frequencies and adds the nonlinear distortion at these low frequencies. The difference between the nonlinear output transformer with and without the hysteresis effect was very small. All these output transformer models – the linear, the nonlinear without the hysteresis and the nonlinear with the hysteresis – were compared using the subjective testing. Two guitar riffs were processed by the simulation of the power amplifier with all transformer models. Output signals for different output transformer models were compared using the ABX listening test, considering the same statistical hypothesis  $H_0$  and  $H_1$ . Each listener had to compare four triplets A,B,X with the linear and the nonlinear transformer model and four triplets with the transformer model with and without the hysteresis. The number of correct identifications needed to reject the null hypothesis  $H_0$  is 36 of total 60. Results of the listening test are given in Table 6.3. As it can be seen, the number of correct identifications is not enough to reject the null hypothesis and therefore, in this case, all transformer models lead to the same audio perception. Because the linear transformer model is much simpler than the other models, it is advantageous to use this model. But of course, this conclusion is valid only for the used transformer model Fender NSC041318.

## 6.5 Audible Aliasing Distortion

The last listening test was focused on the audible aliasing distortion. It is known that the generation of aliasing distortion is the consequence of the nonlinear processing

Table 6.3: Results of transformer model listening test.

Model comparison	Correct identifications	Difficulty to judge
Linear x Nonlinear	32	4.70
Hysteresis x Without hysteresis	31	4.88

of audio signals. The amount of aliasing distortion can be reduced by use of high sampling frequencies and oversampling techniques. The situation is especially critical if the full-bandwidth audio signal is processed. This is not however the case of the guitar signal processing because the frequency bandwidth of the electric guitar is limited. The listening test is designed to determine the over-sampling factor which is needed to reduce the aliasing distortion to inaudible level. A recorded guitar solo served as the input signal for the listening test. The nonlinear distortion was applied on the input signal together with the oversampling technique with different oversampling factors. Listeners were asked to determine if the tested signal is the same as the reference signal. The reference signal was the same distorted guitar signal computed with the oversampling factor equal to 64.

Five guitar signals with different amount of the amplification before the nonlinear processing were used for the test. The amplification was 20, 30, 40, 50 and 60 dB. The nonlinear function was  $y = \tanh(x)$ . The Total Harmonic Distortion (THD) is computed for each amplification using the sine wave signal to identify each guitar riff. Results of the test are in Table 6.4.

Table 6.4: Results of audible aliasing distortion listening test.

THD	Average oversampling factor	Standard deviation	Max. oversampling factor
0.17	2.79	1.97	6
0.36	3.07	2.43	6
0.44	3.29	3.02	10
0.45	4.21	3.62	12
0.46	3.57	4.20	14

The  $4 \times$  oversampling is sufficient for the most of the people but some musicians can hear aliasing distortion until the  $14 \times$  oversampling is used for highly distorted signals. This fact confirms that aliasing distortion was one of the artefacts which

were audible in the listening test for comparing the simulation of the preamp with the hardware device. Note that the last three guitar signals were very highly distorted and that the aliasing distortion was heard only by two people. Therefore,  $8 \times$  oversampling should be sufficient for the most of musicians without hearing the aliasing distortion.

## 6.6 Summary

The Engl E530 guitar preamp was chosen as a case study. It was properly measured and the simulation algorithm was designed using the circuit schematic and measured data of installed tubes. The quality of the simulation algorithm was evaluated using set of testing signals and real guitar samples. The test signals were used for the proper designing of the algorithm and the guitar riffs for the subjective evaluation of the quality. Results showed that the proposed simulation model offers very good quality of the simulation. Although some listeners were able to differ the simulation algorithm and the real device, the most of the listeners were guessing during the listening test.

This model can be approximated using the approximation of triode stages which are connected using the connection current technique. This algorithm is then able to work in real-time with the same accuracy. The amount of data which is used for the approximation can be reduced by the designed algorithm 2 with the maximal error  $7 \times 10^{-4}$  mA.

One of the listening tests was focused on the audible aliasing distortion in the output signal. It was found that use of  $8 \times$  oversampling should be sufficient for the most of musicians but  $4 \times$  oversampling is sufficient on average.

## 7 CONCLUSION

This thesis was focused on the real-time simulation of analog guitar amplifiers and audio effects circuits. This problematic is quite extensive and complex and covers many different fields – the analog circuits analysis (electric and magnetic), the digital signal processing, the computer science etc.

The main aim of the thesis was to design algorithms for the digital signal processing which emulate analog audio circuits with sufficient accuracy but they are still able to work in real-time. While the constraint of working in real-time is given by the computational complexity and can be unambiguously derived from the number of mathematical operations and the computational power of the processor (either DSP or common processor installed in personal computers), the term accuracy of the simulation could be little bit ambiguous. The ideal situation is to produce exactly the same output signal from the simulation algorithm and the real device. This however requires to use very complex model of simulation with computational complexity much higher than what is currently possible to compute in real-time. Nevertheless, the term accurate emulation can also stand for the state when output signals from the simulation algorithm and the real device differ in some details but the audio perception is the same. The model can be then simpler but it must be verified by listening tests.

This work was split into five main chapters. The first one was detailed overview of existing algorithms.. The following chapters were focused on the simulation of simple audio circuits with circuit components which were not discuss in detail earlier. The chapter five was the crucial chapter of this thesis and it dealt with the simulation of more complex systems. The last chapter was focused on the quality of the simulation of analog audio circuits. The quality of simulation was investigated using subjective testing based on listening tests. The results showed that the proposed algorithm offers very good quality of the real-time simulation.

The main contributions of this work are:

- **Incorporation of transformer model into the automated DK-method.**

It allows the automated derivation of DK-method matrices from incidence matrices for circuits containing the transformer (e.g. input stages of some audio effects or the tube push-pull power amplifier). Further, four types of transformer models were compared with regards to the computational complexity, the stability of solution, the accuracy and the audio perception. Modeling the hysteresis effect significantly increases the computational cost and it can introduce the numerical instability but the impact of the hysteresis effect to the output signal is very subtle. It is therefore sufficient to use the GC-model without the hysteresis or the linear model of transformer core whose

use is still sufficient, without introducing audible difference in the output signal, in some circuits.

- **Incorporation of operational amplifier model into the automated DK-method.** It allows the automated derivation of DK-method matrices from incidence matrices for circuits containing operational amplifiers. Two types of operational amplifier models were used – the linear, which is suitable for simulation of analog filter circuits, and the nonlinear, which can be used in simulations of circuits with comparators, e.g. LFO generators.
- **Non-uniformly gridded interpolation.** The efficient interpolation with non-uniformly gridded data and the direct computation of the interpolation interval was introduced. Further, the algorithm for reduction of stored data was designed. This allows the significant reduction of look-up tables sizes.
- **Method of the direct approximation of nonlinear ODEs.** This algorithm provides very efficient simulation of dynamic nonlinear systems which have order of the system (number of states) lower than the number of nonlinear components (nonlinear equations) in the system. The big advantage is that stored data in look-up tables are independent on the sampling frequency value.
- **Modified block-wise method.** This method was designed for the decomposition into simpler blocks in such way that the mutual interaction between adjoined blocks is preserved. This method however assumes that there is no interaction between the first block and third block connected to the second block as the nonlinear load. This assumption is valid e.g. for guitar tube preamps.
- **Block decomposition of the DK-method nonlinear core.** This method enables simulation of the whole circuit without a prior decomposition into blocks. Only the DK-method nonlinear equation can be decomposed into separate equations in case that the simulated circuit has feed-forward topology. This method also assumes very small dependence between the first and the third nonlinear circuit component (connected in series). This dependence can be revealed using the correlation analysis.
- **Block decomposition using connection currents.** This decomposition also enables the simulation of the whole circuit at once. But comparing to the previous one, it is suitable also for feedback systems. DK-method nonlinear equations are split using the unknown connection current. Each simple block, typically of dimension of two, can be easily approximated. Then only the unknown connection currents are computed. The disadvantage is that the connections currents have to be computed numerically. On the other hand, results of this method are comparable to the numerical solving of the whole circuit while the number of unknown variables to be solved is much lower and



thus the computational cost is also much lower.

All proposed algorithms can find their employment in several applications for musicians and some of them have already been used in the software for guitar real-time processing <sup>1</sup>. Listening test showed that very high level of quality of the emulation was achieved. However, there are still some nuances which can be heard. Measurements also showed that the static model of the tube is not sufficiently accurate. The behavior of the grid current is different in the dynamic and the static mode. This model could be improved in future work.

---

<sup>1</sup><http://www.audiffex.com/EN/amplion.html>

## AUTHOR'S PUBLICATIONS

- [1] J. Mačák, “Simulation of the diode limiter using linear time variable filter,” in *Proceedings of the 15th Conference STUDENT EEICT*, Brno, Czech Republic, 2009, pp. 264–268.
- [2] —, “Návrh algoritmů číslicového zpracování zvukových signálů pro simulaci kytarového komba,” in *Proceedings of the 14th Conference STUDENT EEICT*, Brno, Czech Republic, 2008, pp. 73–75.
- [3] J. Mačák and J. Schimmel, “Nonlinear circuit simulation using time-variant filter,” in *Proceedings of the 12th International Conference on Digital Audio Effects DAFx09*, Como, Italy, Sep. 1-4, 2009, pp. 1–5.
- [4] J. Mačák, “Nonlinear transformer simulation for real- time digital audio signal processing,” in *Proceedings of 34th International Conference on Telecommunications and Signal Processing - TSP 2011*, Budapest, Hungary, Aug. 18-20, 2011.
- [5] —, “Nonlinear audio transformer simulation using approximation of differential equations,” *Elektrorevue - Internet journal* (<http://www.elektrorevue.cz>), vol. 2011, no. 4, pp. 22–29, 2011.
- [6] J. Mačák and J. Schimmel, “Simulation of a vacuum-tube push-pull guitar power amplifier,” in *Proceedings of the 14th International Conference on Digital Audio Effects DAFx11*, Paris, France, Sept. 19–23,, 2011.
- [7] J. Mačák and V. Tichý, “Využití interpolace pro simulaci elektroakustických měničů v reálném čase,” *Elektrorevue - Internet journal* (<http://www.elektrorevue.cz>), vol. 2011, no. 25, pp. 1–8, 2011.
- [8] J. Mačák and J. Schimmel, “Real-time guitar tube amplifier simulation using approximation of differential equations,” in *Proceedings of the 13th International Conference on Digital Audio Effects DAFx10*, Graz, Austria, Sep. 6-10, 2010.
- [9] J. Mačák, “Modified blockwise method for simulation of guitar tube amplifiers,” in *Proceedings of 33rd International Conference Telecommunications and Signal Processing TSP - 2010*, Baden near Vienna, Austria, Aug. 17-20, 2010, pp. 1–4.
- [10] —, “Verification of blockwise method for simulation of guitar amplifiers on a guitar tube preamp,” *Elektrorevue* (<http://www.elektrorevue.cz>), vol. 1, no. 2, pp. 16–21, 2010.

- [11] J. Mačák and J. Schimmel, “Real-time guitar preamp simulation using modified blockwise method and approximations,” *EURASIP Journal on Advances in Signal Processing*, vol. 2011, p. 11, 2011.
- [12] J. Mačák, M. Holters, and J. Schimmel, “Simulation of a vacuum-tube push-pull guitar power amplifier,” in *Proceedings of the 15th International Conference on Digital Audio Effects DAFx12*, York, United Kingdom, Sept. 17–21., 2012.

## BIBLIOGRAPHY

- [13] L. W. Nagel and D. Pederson, “Spice (simulation program with integrated circuit emphasis),” EECS Department, University of California, Berkeley, Tech. Rep. UCB/ERL M382, Apr 1973. [Online]. Available: <http://www.eecs.berkeley.edu/Pubs/TechRpts/1973/22871.html>
- [14] U. Zölzer, *DAFX - Digital Audio Effects*, 2nd ed. New York: J. Wiley & Sons, Ltd, 2011.
- [15] J. Schimmel, “Syntéza zvukových efektů s využitím nelineárního zpracování signálů,” Ph.D. dissertation, Brno University of Technology, Brno, 2006.
- [16] —, “Objective evaluation of audible aliasing distortion in digital audio synthesis,” in *Proceedings of 34th International Conference on Telecommunications and Signal Processing TSP*, Budapest, Hungary, Aug. 18–20, 2011, pp. 343–347.
- [17] J. Pakarinen and M. Karjalainen, “Enhanced wave digital triode model for real-time tube amplifier emulation,” *IEEE Transactions on Audio, Speech & Language Processing*, vol. 18, no. 4, pp. 738–746, 2010.
- [18] J. Pakarinen and D. T. Yeh, “A review of digital techniques for modeling vacuum-tube guitar amplifiers,” *Computer Music J.*, vol. 33, no. 2, pp. 85–100, Jun. 2009.
- [19] D. V. Curtis, C. K. Lance, and C. C. Adams, “Simulated tone stack for electric guitar,” Patent 6 222 110, April, 2001. [Online]. Available: <http://www.freepatentsonline.com/6222110.html>
- [20] D. T. Yeh, J. S. Abel, and J. O. Smith, “Automated physical modeling of nonlinear audio circuits for real-time audio effects - part i: Theoretical development,” *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 18, no. 4, pp. 728–737, May 2010.
- [21] D. Yeh, “Automated physical modeling of nonlinear audio circuits for real-time audio effects - Part II: BJT and vacuum tube examples,” *IEEE Trans. Audio, Speech, and Language Processing*, vol. 20, no. 4, pp. 1207–1216, may 2012.
- [22] D. T. Yeh, “Digital implementation of musical distortion circuits by analysis and simulation,” Ph.D. dissertation, Center for Computer Research in Music and Acoustics, Stanford, 2009.

- [23] D. Biolek, “Novel model library of snap 2.6x freeware for analysis of (not only) current-mode circuits,” *Elektrorevue - Internet journal* (<http://www.elektrorevue.cz>), vol. 2004, no. 44, 204.
- [24] D. T. Yeh, J. S. Abel, and J. O. Smith, “Simulation of the diode limiter in guitar distortion circuits by numerical solution of ordinary differential equations,” in *Proc. Digital Audio Effects (DAFx-07)*, Bordeaux, France, Sep. 10-15, 2007, pp. 197–204.
- [25] D. T. Yeh, J. S. Abel, A. Vladimirescu, and J. O. Smith, “Numerical methods for simulation of guitar distortion circuits,” *Computer Music Journal*, vol. 32, no. 2, pp. 23–42, 2008. [Online]. Available: <http://www.mitpressjournals.org/doi/abs/10.1162/comj.2008.32.2.23>
- [26] A. Huovilainen, “Nonlinear digital implementation of the moog ladder filter,” in *Proc. Digital Audio Effects (DAFx-04)*, Naples, Italy, Sep. 5-8, 2004, pp. 61–64.
- [27] R. Melville, L. Trajkovic, S.-C. Fang, and L. Watson, “Artificial parameter homotopy methods for the dc operating point problem,” *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 12, no. 6, pp. 861–877, jun 1993.
- [28] J. O. Smith, *Physical Audio Signal Processing*, 1st ed. USA: W3K Publishing, 2010.
- [29] S. Möller, M. Gromowski, and U. Zölzer, “A measurement technique for highly nonlinear transfer functions,” in *Proc. Digital Audio Effects (DAFx-02)*, Hamburg, Germany, Sep. 26-28, 2002, pp. 203–206.
- [30] M. Karjalainen and J. Pakarinen, “Wave digital simulation of a vacuum-tube amplifier,” in *Proc. Intl. Conf. on Acoustics, Speech, and Signal Proc.*, Toulouse, France, May 15-19, 2006, pp. 153–156.
- [31] J. Pakarinen, M. Tikander, and M. Karjalainen, “Wave digital modeling of the output chain of a vacuum-tube amplifier,” in *Proc. of the Int. Conf. on Digital Audio Effects (DAFx-09)*, Como, Italy, Sept. 1–4, 2009, pp. 1–4.
- [32] D. T. Yeh, J. S. Abel, and J. O. Smith, “Simplified, physically-informed models of distortion and overdrive guitar effects pedal,” in *Proc. Digital Audio Effects (DAFx-07)*, Bordeaux, France, Sep. 10-15, 2007, pp. 189–196.

- [33] D. T. Yeh and J. O. Smith, “Discretization of the ’59 Fender Bassman tone stack,” in *Proc. of the Int. Conf. on Digital Audio Effects (DAFx-06)*, Montreal, Quebec, Canada, Sept. 18–20, 2006, pp. 1–4.
- [34] J. Mačák, “Návrh algoritmů číslicového zpracování signálů pro simulaci kytarových zesilovačů založených na obvodové analýze analogových prototypů,” Master’s thesis, Brno University of Technology, Brno, 2008.
- [35] G. Borin, G. De Poli, and D. Rocchesso, “Elimination of delay-free loops in discrete-time models of nonlinear acoustic systems,” in *Applications of Signal Processing to Audio and Acoustics, 1997. 1997 IEEE ASSP Workshop on*, oct 1997, p. 4 pp.
- [36] A. Fettweis, “Wave digital filters: Theory and practice,” *Proceedings of the IEEE*, vol. 74, no. 2, pp. 270 – 327, feb. 1986.
- [37] K. Meerkotter and R. Scholz, “Digital simulation of nonlinear circuits by wave digital filter principles,” in *Circuits and Systems, 1989., IEEE International Symposium on*, may 1989, pp. 720 –723 vol.1.
- [38] T. Felderhoff, “A new wave description for nonlinear elements,” in *Circuits and Systems, 1996. ISCAS ’96., Connecting the World., 1996 IEEE International Symposium on*, vol. 3, may 1996, pp. 221 –224 vol.3.
- [39] A. Sarti and G. De Poli, “Toward nonlinear wave digital filters,” *Signal Processing, IEEE Transactions on*, vol. 47, no. 6, pp. 1654 –1668, jun 1999.
- [40] D. T. Yeh and J. O. Smith, “Simulating guitar distortion circuits using wave digital and nonlinear state-space formulations,” in *Proc. Digital Audio Effects (DAFx-08)*, Espoo, Finland, Sept. 1-4, 2008, pp. 19–26.
- [41] A. Sarti and G. De Sanctis, “Systematic methods for the implementation of nonlinear wave-digital structures,” *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 56, no. 2, pp. 460 –472, feb. 2009.
- [42] R. Rabenstein, S. Petrausch, A. Sarti, G. De Sanctis, C. Erkut, and M. Karjalainen, “Blocked-based physical modeling for digital sound synthesis,” *Signal Processing Magazine, IEEE*, vol. 24, no. 2, pp. 42 –54, march 2007.
- [43] M. Karjalainen, “Blockcompiler: Efficient simulation of acoustic and audio systems,” in *Proceedings of the 114th AES Convention*, Amsterdam, Netherlands, March 22-25, 2003.

- [44] —, “A research tool for physical modeling and dsp,” in *Proceedings of the 6th International Conference on Digital Audio Effects (DAFx-03)*, London, UK, Sept. 8–11, 2003, pp. 264–269.
- [45] S. Petrausch and R. Rabenstein, “Wave digital filters with multiple nonlinearities,” in *Proceedings of the XII European Signal Processing Conference (EUSIPCO)*, Vienna, Austria, Sept. 2004, p. 77–80.
- [46] R. C. D. de Paiva, J. Pakarinen, V. Välimäki, and M. Tikander, “Real-time audio transformer emulation for virtual tube amplifiers,” *EURASIP Journal on Advances in Signal Processing*, vol. 2011, p. 15, 2011.
- [47] W. J. Rugh, *Nonlinear System Theory: The Volterra/Wiener Approach*, 1st ed. USA: The Johns Hopkins University Press, 1981.
- [48] D. Kocur, *Adaptívne Volterrove číslicové filtre*. Košice: Elfa, s.r.o., 2001.
- [49] T. Hélie, “On the use of Volterra series for efficient real-time simulations of weakly nonlinear analog audio devices: Application to the Moog ladder filter,” in *Proc. of the Int. Conf. on Digital Audio Effects (DAFx-06)*, Montreal, Quebec, Canada, Sept. 18–20, 2006, pp. 7–12, [http://www.dafx.ca/proceedings/papers/p\\_007.pdf](http://www.dafx.ca/proceedings/papers/p_007.pdf).
- [50] A. Novák, “Identification of nonlinear systems: Volterra series simplification,” *Acta Polytechnica Journal of Advanced Engineering*, vol. 47, no. 4-5, May 2007.
- [51] A. Farina, “Nonlinear convolution: A new approach for the auralization of distorting systems,” in *Proceedings of the 100th AES Convection*, Amsterdam, Netherlands, 2001, pp. 12–15.
- [52] A. Novák, S. Lotton, and F. Kadlec, “Modeling of nonlinear audio systems using swept-sine signals: Application to audio effects,” in *Proceedings of the 12th International Conference on Digital Audio Effects DAFX09*, Como, Italy, Sep. 1-4, 2009, pp. 1–6.
- [53] A. Novák, L. Simon, P. Lotton, and J. Gilbert, “Chebyshev model and synchronized swept sine method in nonlinear audio effect modeling,” in *Proceedings of the 13th International Conference on Digital Audio Effects DAFX10*, Graz, Austria, Sep. 6-10, 2010.
- [54] —, “Analysis, synthesis, and classification of nonlinear systems using synchronized swept-sine method for audio effects,” *EURASIP Journal on Advances in Signal Processing*, vol. 2010, p. 8, 2010.

- [55] J. Pakarinen, V. Välimäki, F. Fontana, V. Lazzrini, and J. S. Abel, “Recent advances in real-time musical effects, synthesis, and virtual analog models,” *EURASIP Journal on Advances in Signal Processing*, vol. 2011, p. 15, 2011.
- [56] J. Parker, “A simple digital model of the diode-based ring-modulator,” in *Proc. Digital Audio Effects (DAFx-11)*, Paris, France, Sept. 19–23, 2011.
- [57] C. Raffel and J. Smith, “Practical modeling of bucket-brigade device circuits,” in *Proceedings of the 13th International Conference on Digital Audio Effects DAFx10*, Graz, Austria, Sep. 6-10, 2010.
- [58] J. Timoney, V. Lazarini, A. Gibney, and J. Pekonen, “Digital emulation of distortion effects by wave and phase shaping methods,” in *Proceedings of the 13th International Conference on Digital Audio Effects DAFx10*, Graz, Austria, Sep. 6-10, 2010.
- [59] J. Pekonen, “Coefficient-modulated first order allpass filter as a distortion effect,” in *Proc. Digital Audio Effects (DAFx-08)*, Espoo, Finland, Sept. 1-4, 2008.
- [60] R. C. D. de Paiva, J. Pakarinen, and V. Välimäki, “Reduced-complexity modeling of high-order nonlinear audio systems using swept-sine and principal component analysis,” in *Proc. of the AES 45<sup>th</sup> International Conference: Applications of Time-Frequency Processing in Audio*, Helsinki, Finland, March. 1-4, 2012.
- [61] M. Fink and R. Rabenstein, “A csound opcode for a triode stage of a vacuum tube amplifier,” in *Proc. Digital Audio Effects (DAFx-11)*, Paris, France, Sept. 19–23, 2011.
- [62] G. De Sanctis and A. Sarti, “Virtual analog modeling in the wave-digital domain,” *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 18, no. 4, pp. 715–727, may 2010.
- [63] I. Cohen and T. Helie, “Real-time simulation of a guitar power amplifier,” in *Proceedings of the 13th International Conference on Digital Audio Effects DAFx10*, Graz, Austria, Sep. 6-10, 2010.
- [64] —, “Simulation of a guitar amplifier stage for several triode models: Examination of some relevant phenomena and choice of adapted numerical schemes,” in *Proceedings of the 127th Convention of Audio Engineering Society*, New York, USA, Oct. 9-12, 2009.
- [65] —, “Measures and parameter estimation of triodes, for the real-time simulation of a multi-stage guitar preamplifier,” in *Proceedings of the 129th Convention of Audio Engineering Society*, San Francisco, USA, Nov. 4-7, 2010.



- [66] M. Holters and U. Zölzer, “Physical modelling of a wah-wah effect pedal as a case study for application of the nodal dk method to circuits with variable parts,” in *Proceedings of the 14th International Conference on Digital Audio Effects DAFx11*, Paris, France, Sept. 19–23,, 2011.
- [67] K. Dempwolf, M. Holters, and U. Zölzer, “Discretization of parametric analog circuits for real-time simulations,” in *Proceedings of the 13th International Conference on Digital Audio Effects DAFx10*, Graz, Austria, Sep. 6-10, 2010.
- [68] A. Vlasimirescu, *The Spice Book*, 1st ed. USA: John Wiley & Sons, Inc., 1994.
- [69] K. Dempwolf and U. Zölzer, “A physically-motivated triode model for circuit simulations,” in *Proc. Digital Audio Effects (DAFx-11)*, Paris, France, Sept. 19–23, 2011.
- [70] N. Koren, “Improved Vacuum Tube Models for SPICE Simulations,” 2003, [online], Available from [http://www.normankoren.com/Audio/Tubemodspice\\_article.html](http://www.normankoren.com/Audio/Tubemodspice_article.html).
- [71] T. Serafini, “A Complete Model of a Tube Amplifier Stage,” 2002, [online], Available from <http://www.simulanalog.org/tubestage.pdf>.
- [72] G. Cardarilli, M. Re, and L. Di Carlo, “Improved large-signal model for vacuum triodes,” in *Circuits and Systems, 2009. ISCAS 2009. IEEE International Symposium on*, may 2009, pp. 3006 –3009.
- [73] J. M. Miller, “Dependence of the input impedance of a three-electrode vacuum tube upon the load in the plate circuit,” in *Scientific Papers of the Bureau of Standards*, Washington, USA, Sept. 18–20, 1920, pp. 367–385.
- [74] S. E. Zocholl, A. Guzman, and D. Hou, “Transformer modeling as applied to differential protection,” Schweitzer Engineering Laboratories, Inc. Pullman, Washington, Tech. Rep., 1999.
- [75] E. Sarospataki and M. Kuczmann, “Realization of the jiles-atherton hysteresis model applying the labview and matlab software package,” *Electrical Engineering*, vol. 57, pp. 40–43, August 2006.
- [76] C. Wong, “A dynamic hysteresis model,” *IEEE Transactions on Magnetics*, vol. 24, pp. 1966–1968, MArch 1988.
- [77] M. L. Hodgdon, “Applications of a theory of ferromagnetic hysteresis,” *IEEE Transactions on Magnetics*, vol. 24, no. 1, pp. 218–221, January 1988.

- [78] ———, “Mathematical theory and calculations of magnetic hysteresis curves,” *IEEE Transactions on Magnetics*, vol. 24, no. 6, pp. 218–221, November 1988.
- [79] S. R. Naidu, “Simulation of the hysteresis phenomenon using preisach’s theory,” *IEEE Proceedings*, vol. 137, no. 2, pp. 73–79, March 1990.
- [80] K. H. Carpenter, “A differential equation approach to minor loops in the jiles-atherton hysteresis model,” *IEEE Transactions on Magnetics*, vol. 27, no. 6, pp. 4404–4406, November 1991.
- [81] D. C. Jiles and D. L. Atherton, “Ferromagnetic hysteresis,” *IEEE Transactions on Magnetics*, vol. 19, no. 5, pp. 2183–2185, 1983.
- [82] P. Kis, “Jiles-atherton model implementation to edge finite element method,” Ph.D. dissertation, Budapest University of Technology and Economics, 2010.
- [83] D. C. Jiles, J. B. Thoelke, and M. K. Devine, “Numerical determination of hysteresis parameters for the modeling of magnetic properties using the theory of ferromagnetic hysteresis,” *IEEE Transactions on Magnetics*, vol. 28, no. 1, pp. 27–35, 1992.
- [84] D. C. Jiles and J. B. Thoelke, “Theory of ferromagnetic hysteresis: Determination of model parameters from experimental hysteresis loops,” *IEEE Transactions on Magnetics*, vol. 25, no. 5, pp. 3928–3930, September 1989.
- [85] Q. Chen, L. Xu, X. Ruan, S. C. Wong, and C. Tse, “Gyrator-capacitor simulation model of nonlinear magnetic core,” in *Applied Power Electronics Conference and Exposition, 2009. APEC 2009. Twenty-Fourth Annual IEEE*, feb. 2009, pp. 1740–1746.
- [86] D. C. Hamill, “Gyrator-capacitor modeling: a better way of understanding magnetic components,” in *Proceedings of the 9th Annual Applied Power Electronics Conference and Exposition (APEC '94)*, Feb. 1994, pp. 326–332.
- [87] “Schematic Heaven,” 2000, [online], Available from [http://www.webphix.com/schematic20heaven/www.schematichaven.com/index\\_HTML.html](http://www.webphix.com/schematic20heaven/www.schematichaven.com/index_HTML.html).
- [88] L. J., *Věrný zvuk*, 1st ed. Praha: SNTL, 1962.
- [89] Fender Music Instruments Corp., “Reissue 65 Deluxe Reverb Service Manual,” 1996.
- [90] D. G. Manolakis and J. G. Proakis, *Digital Signal Processing: Principles, Algorithms and Applications*, 3<sup>rd</sup> ed. Englewood Cliffs, NJ, USA: Prentice Hall, 1991.

- [91] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C*, 2nd ed. Cambridge: Cambridge University Press, 1992.
- [92] C. D. Boor, *A Practical Guide to Splines*, 1st ed. New York: Springer, 2001.
- [93] R. Bartels, J. Beatty, and B. Barsky, *An Introduction to Splines for Use in Computer Graphics And Geometric Modeling*, ser. Morgan Kaufmann Series in Computer Graphics and Geometric Modeling. Morgan Kaufmann, 1995.
- [94] Intel, “Intel® 64 and IA-32 Architectures Optimization Reference Manual,” 2009.
- [95] Fender Music Instruments Corp., “Reissue 65 Super Reverb Service Manual,” 1996.
- [96] R. Elliot, “Measuring Thiele / Small Loudspeaker Parameters,” 2007, available at <http://sound.westhost.com/tsp.htm>.
- [97] A. Melka, *Základy experimentální psychoakustiky*. Praha: Akademie múzických umění v Praze, 2005.
- [98] K. Dempwolf, M. Holters, and U. Zölzer, “A triode model for guitar amplifier simulation with individual parameter fitting,” in *Proceedings of the 131th AES Convection*, New York, USA, October 2011.
- [99] O. Kröning, “Entwicklung eines systems zur automatisierten messung von elektronenröhren,” Master’s thesis, Helmut-Schmidt-Universität, Hamburg, 2012.
- [100] R. Zezula, “Objektivní a subjektivní metody vyhodnocování kvality vodoznačných audio signálů.” *Elektrorevue - Internet journal* (<http://www.elektrorevue.cz>), vol. 2008, no. 7, pp. 1–7, 2008.
- [101] T. Heitel, “Využití psychoakustického modelu a tranformace typu wavelet packet pro vodoznačení audio signálů.” Master’s thesis, Brno University of Technology, Brno, 2010.
- [102] N. Cvejic and S. T., *Digital Audio Watermarking Techniques and Technologies : Applications and Benchmarks*, 1st ed. New York: IGI Global, 2007.
- [103] J. Schimmel and J. Misurec, “Characteristics of broken-line approximation and its use in distortion audio effects,” in *Proc. Digital Audio Effects (DAFx-10)*, Bordeaux, France, Sept. 10–15, 2007.

## LIST OF APPENDICES

A	Interpolation Techniques Comparison	180
B	Implementation of Interpolation Formulas	182
C	Incidence Matrices for Fender Type Preamp	184
D	Incidence Matrices for Marshall Preamp	185
E	Incidence Matrices for Marshall Preamp with the Decomposition	187
F	K matrix for Marshall Preamp with the Decomposition	189
G	Instructions for Listening Tests	190
H	Answer Form for Listening Tests	192
I	DVD content	195

## A INTERPOLATION TECHNIQUES COMPARISON

- **Linear interpolation** – provides easy and efficient implementation but on the other hand it provides only the  $C^0$  continuity. When it is used for the approximation of a transfer function, it has similar properties as the piece-wise linear transfer function which was investigated in [103]. The consequence of the non-smooth behavior is the unlimited spectrum bandwidth and hence the generation of aliasing distortion in the output signal. If sufficient number of tabulated function values is used, aliasing distortion can have small values and can be therefore masked. The linear interpolation is given by formula

$$f(x) = Af(x_i) + Bf(x_{i+1}) \quad (\text{A.1})$$

where

$$A \equiv \frac{x_{i+1} - x}{x_{i+1} - x_i} \quad (\text{A.2})$$

and

$$B \equiv 1 - A = \frac{x - x_i}{x_{i+1} - x_i}. \quad (\text{A.3})$$

- **Local polynomial interpolation** – According to the theorem about the existence of the unique interpolation polynomial, every given set of nodes  $x_1, x_2, \dots, x_n$  and function values  $f(x_1), f(x_2), \dots, f(x_n)$  can be interpolated using polynomial of degree  $n - 1$ . Therefore given four nodes  $x_{i-1}, x_i, x_{i+1}, x_{i+2}$  in neighbourhood of a point  $x$  can be interpolated by the cubic polynomial. The efficient construction of this polynomial is provided by the Newton polynomial. Coefficients are computed using divided differences defined by

$$f[x_i, x_{i+1}] = \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i} \quad (\text{A.4})$$

according to

$$\begin{aligned} a_0 &= f(x_i) \\ a_1 &= f[x_{i-1}, x_{i+1}] \\ a_2 &= f[x_{i-1}, x_i, x_{i+1}] \\ a_3 &= f[x_{i-1}, x_i, x_{i+1}, x_{i+2}] \end{aligned} \quad (\text{A.5})$$

and the efficient polynomial evaluation is given by the **nested multiplication**

$$f(x) = ((a_3(x - x_{i+1}) + a_2)(x - x_i) + a_1)(x - x_{i-1}) + a_0. \quad (\text{A.6})$$

This approximation provides more smooth behavior than the linear interpolation but still only  $C^0$  class.

- **Hermite cubic interpolation** – was designed to achieve  $C^1$  continuity. Two conditions have to be fulfilled in each node to get  $C^1$  continuity:  $P_i(x_{i+1}) = P_{i+1}(x_{i+1})$  and  $P'_i(x_{i+1}) = P'_{i+1}(x_{i+1})$  where  $P(x)$  is the cubic polynomial in form

$$P_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3.$$

The polynomial coefficients are obtained from

$$a_i = f(x_i) \quad (\text{A.7})$$

$$b_i = s_i \quad (\text{A.8})$$

$$c_i = \frac{3f[x_{i+1}, x_i] - 2s_i - s_{i+1}}{x_{i+1} - x_i} \quad (\text{A.9})$$

$$d_i = \frac{s_i + s_{i+1} - 2f[x_{i+1}, x_i]}{(x_{i+1} - x_i)^2} \quad (\text{A.10})$$

where  $s_i$  and  $s_{i+1}$  are slopes in nodes  $x_i$  and  $x_{i+1}$  which can be computed e.g. using finite differences.

- **Cubic spline interpolation** – is the piece-wise cubic polynomial interpolation with the  $C^2$  continuity. To achieve  $C^2$  continuity, each node has to additionally fulfill the condition  $P_i^{(2)}(x_{i+1}) = P_{i+1}^{(2)}(x_{i+1})$ . In this case, the slopes  $s_i$  have to be computed from

$$\begin{aligned} (x_i - x_{i-1})s_{i-1} + 2(x_{i+1} - x_{i-1})s_i + (x_{i+1} - x_i)s_{i+1} = \\ = \left( \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i} - \frac{f(x_i) - f(x_{i-1})}{x_i - x_{i-1}} \right) \end{aligned} \quad (\text{A.11})$$

for  $i = 0, 1, \dots, N - 1$ , which requires solving of the linear tridiagonal system for each interpolation or the slopes  $s_i$  have to be also stored in the look-up table. If all spline coefficients  $a_i$ ,  $b_i$ ,  $c_i$  and  $d_i$  are stored in the look-up table, then the interpolation computation consists only from

$$f(x) = ((d_i(x - x_i) + c_i)(x - x_i) + b_i)(x - x_i) + a_i \quad (\text{A.12})$$

if the nested multiplication is used.

Another construction of the cubic spline interpolation introduced in [91] is given by interpolation formula

$$f(x) = Af(x_i) + Bf(x_{i+1}) + Cs_i + Ds_{i+1} \quad (\text{A.13})$$

where  $A$ ,  $B$  are given by (A.2), (A.2) and

$$C \equiv \frac{1}{6}(A^3 - A)(x_{i+1} - x_i)^2 \quad (\text{A.14})$$

and

$$D \equiv \frac{1}{6}(B^3 - B)(x_{i+1} - x_i)^2. \quad (\text{A.15})$$

## B IMPLEMENTATION OF INTERPOLATION FORMULAS

Listing B.1 implements the linear interpolation given by (A.1).

Listing B.1: Linear interpolation

```
h = x_breaks [ i+1 ] - x_breaks [ i ];
A = (x_breaks [ i+1]-x)/h;
B = 1.f-A;
return A*fx [ i ] + B*fx [ i+1];
```

The local polynomial interpolation which uses the Newton interpolation polynomial (A.5), (A.6) is in listing B.2

Listing B.2: Local Newton polynomial interpolation

```
a0 = fx [ i-1 ];
a1 = (fx [ i]-fx [ i-1 ])/(x_breaks [ i]-x_breaks [ i-1 ]);
tmp1 = (fx [ i+1]-fx [ i ])/(x_breaks [ i+1]-x_breaks [ i ]);
tmp2 = (fx [ i+2]-fx [ i+1 ])/(x_breaks [ i+2]-x_breaks [ i+1 ]);
a2 = (tmp1-a1)/(x_breaks [ i+1]-x_breaks [ i-1 ]);
tmp3 = (tmp2-tmp1)/(x_breaks [ i+2]-x_breaks [ i ]);
a3 = (tmp3-a2)/(x_breaks [ i+2]-x_breaks [ i-1 ]);
out = a3*(x-x_breaks [ i+1 ]) + a2;
out = out*(x-x_breaks [ i ]) + a1;
return out*(x-x_breaks [ i-1 ]) + a0;
```

Listing B.3 implements the Hermite cubic interpolation given by (A.7), (A.8), (A.9) and (A.10). Slopes are computed using finite differences and in the listing, they are stored in variables D1 and D2.

Listing B.3: Hermite cubic interpolation

```
dx = x_breaks [ i+1 ] - x_breaks [ i ];
df = (fx [ i+1]-fx [ i ])/dx;
D1 = (fx [ i+1]-fx [ i-1 ])/(x_breaks [ i+1 ] - x_breaks [ i-1 ]);
D2 = (fx [ i+2]-fx [ i ])/(x_breaks [ i+2 ] - x_breaks [ i ]);
c = (3*df - 2*D1 -D2)/dx;
d = (-2*df + D1 +D2)/(dx*dx);
x = in - x_breaks [ i ];
out = d*x+c;
out = out*x + D1;
return out*x + fx [ i];
```

Listing B.4 provides the implementation of the cubic spline evaluation (A.12). This however requires the precomputation of spline coefficients, which are stored in vectors `a_vec`, `b_vec`, `c_vec` and `d_vec`.

Listing B.4: Cubic spline interpolation 1

```
x = in - x_breaks[fx];
out = d_vec[i]*x + c_vec[i];
out = out*x + b_vec[i];
return out*x + a_vec[i];
```

Implementation of the cubic spline defined in listing B.5 requires only precomputed coefficients `a_vec` and `c_vec`. The remaining coefficients are computed during the interpolation. The computational cost is therefore higher but on the other hand the memory demands are lower.

Listing B.5: Cubic spline interpolation 2

```
h = x_breaks[i+1] - x_breaks[i];
dh = 1./h;
d = (c_vec[i+1] - c_vec[i])*0.33333*dh;
b = (fx[i+1] - fx[i])*dh - (2.f*c_vec[i] + c_vec[i+1])*h*0.33333;
x = in - x_breaks[i];
out = d_vec[i]*x + c_vec[i];
out = out*x + b_vec[i];
return out*x + fx[i];
```

The last interpolation algorithm in listing B.6 implements the equation (A.13) and also require precomputed function values and the second derivative in vector `c_vec`.

Listing B.6: Cubic spline interpolation 3

```
h = x_breaks[i+1] - x_breaks[i];
A = (x_breaks[i+1]-in)/h;
B = 1.f-A;
lin = A*fx[i] + B*fx[i+1];
return lin + ((A*A*A-A)*c_vec[i] + (B*B*B-B)*c_vec[i+1])*(h*h)/6.f;
```



## C INCIDENCE MATRICES FOR FENDER TYPE PREAMP

$$N_r = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

$$N_x = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

$$N_n = \begin{bmatrix} 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 \end{bmatrix}$$

$$N_u = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

$$N_o = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

$$G_r = \text{diag} \left( \frac{1}{R_1}, \frac{1}{R_2}, \frac{1}{R_3}, \frac{1}{R_4}, \frac{1}{R_5}, \frac{1}{R_{6a}}, \frac{1}{R_{6b}}, \frac{1}{R_7}, \frac{1}{R_8}, \frac{1}{R_{9a}}, \frac{1}{R_{9b}}, \frac{1}{R_{10}}, \frac{1}{R_{11}} \right)$$

$$G_r = \frac{2}{T} \text{diag} (C_1, C_2, C_3, C_4, C_5).$$



$$G_r = \text{diag} \left( \frac{1}{R_1}, \frac{1}{R_2}, \frac{1}{R_3}, \frac{1}{R_4}, \frac{1}{R_5}, \frac{1}{R_{6a}}, \frac{1}{R_{6b}}, \frac{1}{R_7}, \frac{1}{R_8}, \frac{1}{R_9}, \frac{1}{R_{10}}, \frac{1}{R_{11}}, \frac{1}{R_{12}} \right)$$

$$G_r = \frac{2}{T} \text{diag} (C_1, C_2, C_3, C_4, C_5, C_6, C_7, C_{m1}, C_{m2}, C_{m3}).$$



$$N_o = [ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 ]$$

$$G_r = \text{diag} \left( \frac{1}{R_1}, \frac{1}{R_2}, \frac{1}{R_3}, \frac{1}{R_4}, \frac{1}{R_5}, \frac{1}{R_{6a}}, \frac{1}{R_{6b}}, \frac{1}{R_7}, \frac{1}{R_8}, \frac{1}{R_9}, \frac{1}{R_{10}}, \frac{1}{R_{11}}, \frac{1}{R_{12}} \right)$$

$$G_r = \frac{2}{T} \text{diag} (C_1, C_2, C_3, C_4, C_5, C_6, C_7, C_{m1}, C_{m2}, C_{m3}).$$

## F K MATRIX FOR MARSHALL PREAMP WITH THE DECOMPOSITION

$$\mathbf{K} = \begin{bmatrix} 6.69 \times 10^4 & 8.17 \times 10^2 & 0 & 0 & 0 & 0 & 8.02 \times 10^2 & 0 & 0 \\ 8.17 \times 10^2 & 5.04 \times 10^4 & 0 & 0 & 0 & 0 & 5.04 \times 10^4 & 0 & 0 \\ 0 & 0 & 8.14 \times 10^5 & 2.88 \times 10^4 & 0 & 0 & -8.04 \times 10^5 & 1.88 \times 10^4 & 0 \\ 0 & 0 & 2.88 \times 10^4 & 1.08 \times 10^5 & 0 & 0 & -1.88 \times 10^4 & 9.81 \times 10^4 & 0 \\ 0 & 0 & 0 & 0 & 4.24 \times 10^5 & 1.07 \times 10^4 & 0 & -4.23 \times 10^5 & 0 \\ 0 & 0 & 0 & 0 & 1.07 \times 10^4 & 9.87 \times 10^4 & 0 & -9.92 \times 10^3 & 0 \\ 8.02 \times 10^2 & 5.04 \times 10^4 & -8.04 \times 10^5 & -1.88 \times 10^4 & 0 & 0 & 8.77 \times 10^5 & -1.88 \times 10^4 & 0 \\ 0 & 0 & 1.88 \times 10^4 & 9.81 \times 10^4 & -4.23 \times 10^5 & -9.92 \times 10^3 & -1.88 \times 10^4 & 5.43 \times 10^5 & 0 \end{bmatrix}$$

## G INSTRUCTIONS FOR LISTENING TESTS

Dear participants,

Hereby I would like to ask you to take part in the listening test. The aim of the test is to determine the quality of the digital simulation of analog effects. Results of this test will be used in my dissertation thesis. Results will be anonymous and you do not have to be worried that your answers will be published. Evaluation of your abilities is not intention of this test.

The listening test consists of several parts which can be divided by small breaks if you need or want. The first part is focused on the determination of quality of the guitar preamp Engl E530 simulation. The ABX listening test will be used. Three sound samples will be available - one is measured, one is simulated (samples A and B) and one is unknown (sample X). Your task is to determine whether the sample X is the sample A or B. You can switch between all three samples during the playback (it will be shown). There are 20 tests which are assembled randomly and which cover different playing styles. 10 tests are focused on the simulation of the clean channel of the preamp and 10 tests are focused on the simulation of the crunch channel of the preamp. There is the prepared answer form for filling in the answers. Please fill in the type of the sample (A or B) into the column Type. Focus only on the guitar signal. If you are not sure with the answer, choose A or B randomly. Further fill in how much it was difficult to distinguish the samples (1 to 5, one for easy and 5 for very difficult) and if you are satisfied with the simulation quality (1 to 5, 1 for insufficient and 5 for sufficient). Do not go back to previous tests.

The second task is to determine if the simplified model of the output transformer has an audible impact to the output signal. The ABX test will be used again.

The third task is the determination of the minimal number of data for the approximation which are required without losing the same audible perception. Reduce the number of data using the slider until you can hear any difference between the "Full" and "Reduced" variant (it will be shown). Write down into the answer form the number of data points required for each type of the approximation.

The last task is to determine if there is a noticeable aliasing distortion in the sound sample. The test consists of 5 differently distorted guitar riffs using the guitar effect Distortion organized in 5 groups. Each group is denoted with a different letter. Each guitar riff from the given group is processed in such way that it consists different amount of the aliasing distortion - number 1 contains the most of aliasing distortion while number 5 less or none aliasing distortion. There is also the reference sample without aliasing distortion to be compared with. Go through all numbers in each group sequentially and determine if you can hear any difference between the sample A and the reference sample B. As soon as you cannot hear the difference,

omit the samples with higher number and continue with the next group.

Please read these instructions carefully and in case of any doubts do not hesitate to ask the experiment leader. Incomprehension of these instructions can lead to wrong results which can devalue this test.

Thank you for your time and cooperation.



# H ANSWER FORM FOR LISTENING TESTS

## Clean channel

Experiment	Type (A, B)	Difficulty	Quality
A1			
A2			
A3			
A4			
A5			
A6			
A7			
A8			
A8			
A9			
A10			

## Crunch channel

Experiment	Type (A, B)	Difficulty	Quality
B1			
B2			
B3			
B4			
B5			
B6			
B7			
B8			
B8			
B9			
B10			

## Output Transformer Model

Experiment	Type (A, B)	Difficulty	Quality
C1			
C2			
C3			
C4			
C5			
C6			
C7			
C8			
C8			

## Aliasing distortion

Experiment	Match
D1	
D2	
D3	
D4	
D5	
D6	
D7	
D8	
D8	
E1	
E2	
E3	
E4	
E5	
E6	
E7	
E8	
E8	
F1	
F2	
F3	
F4	
F5	
F6	
F7	
F8	
F8	
G1	
G2	
G3	
G4	
G5	
G6	
G7	
G8	
G8	
H1	
H2	
H3	
H4	
H5	
H6	
H7	
H8	
H8	

## Approximation parameters

Type	Number of points
Spline 1	
Spline 2	
Hermite	
Newton	
Linear	

# I DVD CONTENT

- **Matlab** – organized according to chapters of the thesis. Each folder contains the README.txt file with a brief description of all files.
  - common – contains functions which are shared by more chapters,
  - chapter 3 – functions and scripts with simulations of transformers and operational amplifiers,
  - chapter 4 – functions and scripts for approximation of functions,
  - chapter 5 – simulations of more complex systems,
  - chapter 6 – functions and scripts for the evaluation of quality of simulation.
- **Listening test files**
  - source wave files which were used for the listening test,
  - VST plug-in effects used in software cubase during the listening test,
  - documents used for the listening test.
- **Applications** – demo version of commercial product AmpLion which use algorithms designed in this thesis.

Curriculum Vitae  
**Jaromír Mačák**

Tel: (+420) 777 297 164

Email: jarda.macak@seznam.cz

## Personal information

- Born on 4th October, 1983.
- Czech nationality

## Education

- Brno University of Technology, Faculty of Electrical Engineering and Communication, Czech Republic, doctoral degree programme, 09/2008 – 10/2012
- Helmut Schmidt University - University of the Federal Armed Forces Hamburg, Department of Electrical Engineering, Hamburg, Germany, 02/2012 – 05/2012  
– research stay
- Brno University of Technology, Faculty of Electrical Engineering and Communication, Czech Republic, master's degree programme, 09/2006 – 06/2008  
– final state exam with honours
- Brno University of Technology, Faculty of Electrical Engineering and Communication, Czech Republic, bachelor's degree programme, 09/2003 – 09/2006  
– final state exam with honours
- Gymnázium Lipník nad Bečvou, 09/1995 – 06/2003  
– final exam with honours

## Employment

- Brno University of Technology, Department of Telecommunication, Brno, 08/2009 – 05/2012  
– assistant on project CZ.1.07/2.3.00/09.0222
- Janáček Academy of Music and Performing Arts in Brno (JAMU in Brno), Brno, 01/2010 – 08/2012  
– specialized assistant and lecturer in field of audio effects and instruments design
- Disk Multimedia, spol. s r.o., Boskovice, 01/2006 – 10/2012, software developer  
– DSP algorithm design

- \* digital simulation of Gallien-Krueger bass amplifiers  
<http://www.audiffex.com/EN/gallien.html>,
- \* digital simulation of TC Electronic vintage guitar pedals  
<http://www.tcelectronic.com/vintageguitarpedalbundle.asp>,
- \* digital simulation of analog guitar amplifiers and effects – ampLion  
<http://www.audiffex.com/EN/amplion.html>,
- \* audio effects for TC Powercore system – inValve  
<http://www.audiffex.com/EN/invalve.html>,
- audio plug-in development
- team-leader of audio effects development
- GUI development

## Participation in Projects

- CZ.1.07/2.3.00/09.0222 – Educational Center for Increasing the Interest of Young People in Research into Information and Communication Technologies. Holder: Ing. Kubánek. 2009–2012
- FR-TI1/495 – Manifold System for Multimedia Digital Signal Processing. Holder: Ing. Schimmel. 2009–2012
- 2704/G1/2011 – Extension of Laboratory Assignments Addressing Audio Effect Measurement. Holder: Ing. Mačák. 2011
- FEKT-S-11-17 – Research of Sophisticated Methods for Digital Audio and Image Signal Processing. Holder: Prof. Z. Smékal. 2011
- MSM21630513 – Electronic Communication Systems and Technologies of Novel Generations (ELKOM). Holders: Prof. Z. Raida, Prof. K. Vrba, Prof. J. Jan. 2008–2011
- 2912/G1/2010 – Innovation of Laboratory Assignments of Subject Electroacoustics. Holder: Ing. Mačák. 2010
- FEKT-S-10-16 – Research on Electronic Communication Systems. Holder: Prof. K. Vrba. 2010
- FT-TA3/010 – Spatial Effects for Multichannel Digital Signal Processing Systems. Holder: Ing. Schimmel. 2009.

## Invited Talks

- Simulation of analog audio devices in real-time. Audio Engineering Society, Prague 1/12/2010.

- Real-time digital simulation of analog audio effects, SPLab Workshop 2011, Brno 28/10/2011.

## Results

- Publications: 13
  - In international journals with Impact Factor: 1
  - In proceedings of international conferences: 6
  - In other journals: 4
  - In other conferences: 2
- Software/Products: 1
- Citations (without self-citations): 8
- h-index according to Web of Science: 1
- responses to publications from foreign experts: 3

## Awards

- EEICT 2008 student conference and competition – 3rd prize
- EEICT 2090 student conference and competition – 1st prize