Automated Physical Modeling of Nonlinear Audio Circuits For Real-Time Audio Effects - Part II: BJT and Vacuum Tube Examples

David T. Yeh, *Member, IEEE*

***Abstract*—This is the second part of a two-part paper that presents a procedural approach to derive nonlinear filters from schematics of audio circuits for the purpose of digitally emulating musical effects circuits in real-time. This work presents the results of applying this physics-based technique to two audio preamplifier circuits. The approach extends a thread of research that uses variable transformation and offline solution of the global nonlinear system. The solution is approximated with multidimensional linear interpolation during runtime to avoid uncertainties in convergence. The methods are evaluated here experimentally against a reference SPICE circuit simulation. The circuits studied here are the BJT common emitter amplifier, and the triode preamplifier. The results suggest the use of function approximation to represent the solved system nonlinearity of the K-method and invite future work along these lines.**

***Index Terms*—**Virtual analog, physical modeling synthesis, guitar distortion, guitar amplifier modeling, vacuum tube amplifier, real-time audio, circuit simulation, nonlinear filters, K-Method, ODE solver, **EDICS: AUD-SYST**

I. Introduction

THIS is the second part of a two-part paper that presents research to model and simulate highly nonlinear circuits used primarily for electric guitar effects. This method in its present form is most directly applicable to circuits without time-varying parameters, such as guitar distortion circuits. Part I [1] presented a procedural approach to derive nonlinear filters from schematics of audio circuits for the purpose of digitally emulating musical effects circuits in real-time. The focus of Part II is the simulation of two circuits common in audio amplifiers: a BJT common emitter amplifier, and a vacuum tube triode preamp.

This effort aims to preserve the heritage of musical circuits whose components, such as vacuum tubes, or vintage transistors, are becoming increasingly rare. By exploiting the progress of contemporary digital computing power, modeling vintage circuits based on archives of their circuit schematics and device characteristics can ensure that the unique sound of these circuits will be available for future generations of musicians.

This work extends an established efficient nonlinear continuous-time state-space formulation for physical modeling of musical acoustics to solve in real-time the ordinary differential equations (ODEs) of nonlinear circuits. Software was developed that accepts circuits described in netlist form and given user design parameters, generates a real-time nonlinear circuit simulator. This software, sound examples with various input signals and real-time effect plugins are available at http://ccrma.stanford.edu/~dtyeh/nkmethod10. The plugins were prototyped using the LV2 audio effects framework[1] using open-source audio and math libraries.

A review of related work in real-time models of musical circuits and a brief summary of this method are presented, followed by the model and simulation results of two example circuits.

The author is affiliated with the Center for Computer Research in Music and Acoustics (CCRMA), Department of Music, Stanford University, Stanford, CA 94305-8180, USA. This work was partially supported by a National Science Foundation Graduate Fellowship.

II. Previous Work

A variety of methods for digital emulation of nonlinear musical circuits have been attempted through the years [2], [3]. Most commercial approaches rely on some combination of waveshaping with filtering, which is a simplified approach to simulating the nonlinear behavior of the circuit.

Nonlinear processing multiplies the bandwidth of the input signal and in discrete-time processing these frequency multiples can alias at the Nyquist frequency. Consequently, most musical distortion algorithms apply the nonlinearity at an upsampled rate, typically 4 to 8 times the audio sampling rate, e.g., 8x48 kHz. Furthermore, the baseband tones produce psychoacoustic masking patterns in human hearing that can hide the weaker alias tones even using lower oversampling factors as studied in [4].

Physically-based methods numerically integrate the ordinary differential equations of the circuit. The explicit integration methods have poor convergence properties, which depend upon the input signal, for highly saturating circuits such as guitar amplifiers (see [5] for a summary of integration methods). Because of this, recent attempts to emulate nonlinear musical circuits have used implicit numerical integration [1], [4], [6]–[8]. Wave digital filter [4], [6] and time-varying filter methods [7] are also essentially implicit numerical integration schemes. The nonlinear solver in implicit methods has difficultly converging when applied to circuits with strongly saturating nonlinearities [5]. A workaround is to solve the nonlinear functions offline and use a function approximation such as table lookup during runtime [9], [10].

[1]http://lv2plug.in

One popular application of virtual analog is to simulate the nonlinearities and dynamics added by amplification with a vacuum tube amplifier. This is commonly done using wave-shaping [2]. Accurately simulating the tube sound based on physical principles in real-time is a challenging task. Wave digital models of the vacuum tube amplifier require certain approximations and modifications that affect the stability and dynamics of the system [4], [6], [11]. The wave digital approach is not a straightforward method to simulate arbitrary circuits although it does offer advantages in numerical stability and computational efficiency when used properly [10], [12].

Recent efforts [13]–[15] are converging toward direct solution of the differential equations with optional precomputation and tabulation of the system nonlinearity to mitigate the convergence problem, which is discussed in Part I [1]. The technique of table lookups to represent nonlinearities has been used for decades in digital audio [16], [17].

One form of this approach, the K-method [18] – a nonlinear, state-space system solver – can be used to simulate tube amps [10] without many of the approximations necessary in wave digital and direct ODE approaches. The K-method itself is a special form of the general method to compute a system of connected nonlinear filters [19], [20] by rearranging the system and solving a system-level nonlinear equation using table lookup. Owing to their generality, these methods may have numerous formulations to model the same circuit, and are laborious to apply.

The DK-method presented in Part I of this paper [1] makes several contributions to extend this research. It provides a set of unambiguous rules to map the circuit features to state variables and the system matrices. The specific formulation proposed preserves the dimensionality of the underlying nonlinearity, whereas the nonlinear filter composition method does not dictate how to represent the solved system nonlinearity. Because it preserves the system inputs and components, relative to solving the ODE directly, this method is demonstrated to have the built-in ability to model details such as modulation caused by the power supply voltage. Finally the procedure is suited to implementation as software so that the user does not need to derive circuit equations by hand. For mildly distorting circuits, the nonlinearity can be solved online even for large circuits. For strongly clipping circuits, the nonlinearity must be computed offline. In the present implementation, this nonlinearity is represented as a lookup table, which limits the size of a circuit that can be solved automatically. Automated analysis of a circuit to determine suitable partitioning would also be possible [21], perhaps with some user interaction, but is left for future work.

III. Discrete K-Method

The Discrete K-method (DK-method) is briefly reviewed here before it is applied to the example circuits. The DK-method starts by scanning the netlist and building a matrix system of equations using a set of templates given in [1] based on modified nodal analysis (MNA) [22]. The templates describe how each circuit element type contributes to the system matrices. This system is solved to find state transition coefficient matrices and the nonlinear relationships between system variables. This produces a form of nonlinear filter similar to that of the K-method. Whereas the K-method requires derivation of the system ODE before discretization, the DK-method first applies discretization, then solves the resulting system for the state update and nonlinear relations. This ordering of operations simplifies the automation of the system derivation.

A. Solution of the discrete K-method

Discretizing the circuit and performing MNA on the circuit yields a state update equation in the following form, as detailed in Part I [1]:

$$\mathbf{x}[n] = \mathbf{A}\mathbf{x}[n-1] + \mathbf{B}\mathbf{u}[n] + \mathbf{C}\mathbf{i}[n]. \tag{1}$$

The state is described by vector $\mathbf{x}$; $\mathbf{u}$ is the vector of inputs to the circuit, including power supplies, and $\mathbf{i}$ is the vector contribution of the nonlinear devices in the circuits, i.e., the diodes, vacuum tubes, etc. These nonlinear devices are typically modeled as voltage controlled current sources

$$\mathbf{i} = \mathbf{f}(\mathbf{v}), \tag{2}$$

where $\mathbf{v}$ is the vector of controlling voltages for the nonlinear devices. These controlling voltages are also solved from the circuit equations by MNA and written as a linear combination of $\mathbf{x}$, $\mathbf{u}$, and $\mathbf{i}$.

$$\mathbf{v} = \mathbf{D}\mathbf{x} + \mathbf{E}\mathbf{u} + \mathbf{F}\mathbf{i}, \tag{3}$$

The terminal voltages, $\mathbf{v}[n]$, for the nonlinear devices can be solved implicitly

$$\mathbf{0} = \mathbf{p}[n] + \mathbf{F}\mathbf{f}(\mathbf{v}[n]) - \mathbf{v}[n], \tag{4}$$

$$\mathbf{p}[n] = \mathbf{D}\mathbf{x}[n-1] + \mathbf{E}\mathbf{u}[n], \tag{5}$$

The parameter $\mathbf{p}$ is introduced to rewrite (4) and (2) as explicit mappings

$$\mathbf{v}[n] = \mathbf{\Gamma}(\mathbf{p}[n]), \tag{6}$$

$$\mathbf{i}[n] = \mathbf{f}(\Gamma(\mathbf{p}[n])) \tag{7}$$

which is possible under conditions described in Part I.

Outputs $\mathbf{y}$ from the circuit are also solved using MNA and can be written as a linear combination of $\mathbf{x}$, $\mathbf{u}$, and $\mathbf{i}$:

$$\mathbf{y} = \mathbf{L}\mathbf{x} + \mathbf{M}\mathbf{u} + \mathbf{N}\mathbf{i}. \tag{8}$$

In general, the dimensionality of the parameter $\mathbf{p}$ corresponds to the number of controlling voltage pairs in the system.

B. Multidimensional linear interpolation

In order to circumvent convergence problems at runtime, the nonlinearity (6) was precomputed using Newton homotopy, as described in Part I, and stored as a table for real-time implementations of the nodal K-method. The table was implemented using a uniform grid for constant time accesses. Interpolation of grid points was performed using multidimensional linear interpolation, drawing upon the C++ code implementing `interpn` from the open-source project Octave [23]. The step sizes and extents of the table are design parameters determined experimentally by running the DK-method runtime algorithm to find the extents of the table given an input level, and finding the step size that produces acceptable error in the DK-method output.

C. DK-method runtime algorithm

The runtime loop for this discrete-time K-method can be summarized as

1) Change of variable to parameter using (5),
2) Nonlinear function lookup $\mathbf{i}[n] = \mathbf{f}(\mathbf{\Gamma}(\mathbf{p}[n]))$ using solved system nonlinearity (6), (2),
3) State update using (1),
4) Compute outputs using (8).

This DK-method runtime core is written generically as a matrix operation, taking as parameters the matrix coefficients, nonlinear function approximation, and vector assignments to simulate specific circuits and is shown as a block diagram in Fig. (1).

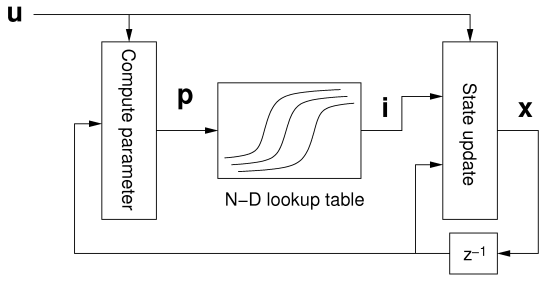

Figure 1. The nonlinear filter algorithm resulting from precomputing the nonlinear part of the ODEs of a circuit.

D. DK-method codegen procedure

The DK-method scans the netlist describing the circuit and builds the MNA system according to a set of templates corresponding to circuit elements. It then solves for the DK-method parameters as derived above, which can be used to generate a recursive nonlinear filter for use in a runtime loop.

In this work, a script written in Python parses the netlist, computes the coefficients, and generates a C++ class that encapsulates the DK-method solver for that circuit. This C++ class is then incorporated into a plugin, where resampling matches the plugin sampling rate to the DK-method sampling rate. For this work we use the LV2 plugin architecture, and the linear interpolation resampler from the libsamplerate library[2].

[2]http://www.mega-nerd.com/SRC

Computing the symbolic expression for the coefficients is done using the Sage Math[3] interface to Maxima[4]. This quickly becomes intractable for systems with more than three state variables and two nonlinear variables, therefore the parser can optionally produce C++ code to compute the DK-method coefficients numerically, which in practice is fast and accurate for the small circuits found in distortion devices. The C++ matrix library Gmm++[5] provides the matrix inverse and LU solver algorithms for this work.

The procedure to model a circuit using the DK-method is as follows

1) Describe circuit netlist and device model parameters,
2) Run the script, which generates a C++ class that extends the DK-method core algorithm with methods to compute the DK-method coefficient matrices for the circuit, and assigning vector entries to system variables,
3) Generate the function approximation of (7), e.g. tabulate the function for lookup and interpolation.
4) Wrap the class with audio I/O code, e.g. as an LV2 plugin.

E. Runtime computational effort

The runtime algorithm computes three matrix equations, and a function approximation for each nonlinear current in the model. For a system with n states, m inputs, ν nonlinear control voltages, and η nonlinear currents, to compute $\mathbf{p}$ using (5) would be a $\nu \times n$ matrix multiply with a vector length n added to a $\nu \times m$ matrix multiply with vector length m. A straightforward implementation would require $\nu\,(n + m)$ multiplies and $\nu(n + m - 1)$ adds.

Likewise, state update of $\mathbf{x}$ requires $n(n+m+\eta)$ multiplies and $n(n + m + \eta - 1)$ adds. Computation of the outputs $\mathbf{y}$ requires $n + m + \eta$ multiplies and $n + m + \eta - 1$ adds.

Linear interpolation in ν dimensions on a uniform grid retrieved from memory is a constant time operation and requires a transformation of the values of ν to table indices, followed by 2^ν memory lookups and $2^\nu - 1$ one-dimensional linear interpolations. A straightforward implementation to transform ν requires 1 multiply, 2 adds, and a `floor` operation. Memory lookups may incur tens of cycles of delay for a cache hit, or hundreds if accessing main memory on contemporary x86 architecture PCs. Denote this memory lookup cost as c. Each 1-D linear interpolation requires 2 multiplies and 3 adds.

Assuming multiplies, adds, and `floor` each take 1 cycle, and memory operations take c cycles, and approximating matrix-vector multiplies as having as many adds as multiplies, the total cycles to compute a sample is

$$2\nu(n+m)+2(n+1)(n+m+\eta)+\eta((4\nu+c2^\nu+5(2^\nu-1)). \quad (9)$$

IV. Simulation of Audio Preamplifiers

The DK-method is validated on standard building blocks found in musical distortion circuits. The following provides context for the common emitter BJT amplifier and the common

[3]http://www.sagemath.org/
[4]http://maxima.sourceforge.net/
[5]http://home.gna.org/getfem/gmm_intro

cathode vacuum tube triode preamplifier. The matrix parameters were derived symbolically and tend to be too complicated to be displayed here; however, the runtime code is available online.

Both examples here involve complicated two-dimensional nonlinearities with three state variables. To evaluate the nodal DK-method in this situation, the Backward Euler (BE) discretization was applied at 8x oversampling of $f_s = 48\,\mathrm{kHz}$. The Newton's method parameters are $\mathrm{RELTOL} = 10^{-4}$, $\mathrm{MAXRES} = 10^{-3}$, as defined in Part I.

To explore the effects of approximation error in the system nonlinearity lookup table, we tested a coarsely sampled table with 25 steps for each dimension. In comparison, we tested a medium table with 100 steps per dimension.

Comparison to SPICE verifies the correctness of our implementation as a sanity check. The same device models were used in LTspice IV (Linear Technology), using default simulation parameters (modified trapezoidal rule integration), a maximum timestep of $2.6\,\mu$s, and saving the output to a wave file at 384 kHz, which applies linear interpolation to fit SPICE's output to a fixed sampling grid.

The 8x oversampled BE algorithm was chosen because it is simple and stable, and at the highly oversampled 8x rate to mitigate aliasing, the numerical integration error compared to LTspice is largely located above the audible frequency range of 20Hz – 20kHz as evinced by the spectral plots in the results.

A. Common Emitter BJT amplifier

Figures 2 and (4) show the common-emitter amplification stage from the Boss DS-1 [24]. It provides roughly 30-dB of preamplification gain as well as some odd-order saturating nonlinearity.

The design values for this circuit are $R_i = 100\,\mathrm{k}\Omega$, $R_c = 10\,\mathrm{k}\Omega$, $R_l = 100\,\mathrm{k}\Omega$, $R_f = 470\,\mathrm{k}\Omega$, $R_e = 22\,\Omega$, $C_i = 0.047\,\mu\mathrm{F}$, $C_f = 250\,p\mathrm{F}$, and $C_o = 0.47\,\mu\mathrm{F}$.

Figure 3 depicts generic model for the bipolar junction transistor comprising voltage-controlled current sources. The BJT has three terminals, the collector, base, and emitter, whose currents are controlled by voltages across two pairs of the terminals, $V_{be} = V_b - V_e$, $V_{bc} = V_b - V_c$. By conservation of current, only two of the terminal current definitions are needed to completely describe the current-voltage (I-V) characteristics. We use $I_b(V_{be}, V_{bc})$ and $I_c(V_{be}, V_{bc})$ here. Semiconductor devices such as the BJT also have nonlinear resistances and capacitors, which require more detailed models; however, for simplicity, we assume that we can neglect these effects for the signal levels of this circuit in the audio frequency band.

A complete, yet simple, physically derived model for computer simulation, the Ebers-Moll model [25] defines the following current-voltage (I-V) characteristics:

$$I_e = \frac{I_S}{\alpha_F}\left[\exp\left(V_{be}/V_T\right) - 1\right] - I_S\left[\exp\left(V_{bc}/V_T\right) - 1\right] \quad (10)$$

$$I_c = I_S\left[\exp\left(V_{be}/V_T\right) - 1\right] - \frac{I_S}{\alpha_R}\left[\exp\left(V_{bc}/V_T\right) - 1\right] \quad (11)$$

$$I_b = \frac{I_S}{\beta_F}\left[\exp\left(V_{be}/V_T\right) - 1\right] + \frac{I_S}{\beta_R}\left[\exp\left(V_{bc}/V_T\right) - 1\right] \quad (12)$$

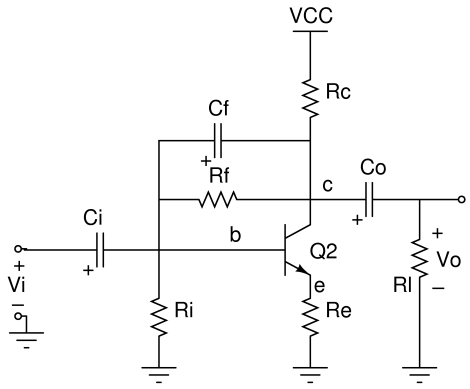

Figure 2. Schematic of the common-emitter amplifier with feedback.

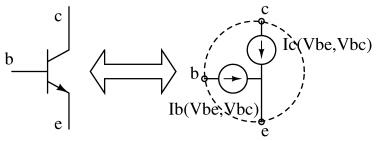

Figure 3. Generic BJT device model.

Device parameters for this simulation are $V_T = 26\,\mathrm{mV}$, $\beta_F = 200$, $\beta_R = 0.1$, $\alpha_R = \beta_R/(1+\beta_R)$, $I_s = 6.734 \times 10^{-15}\mathrm{A}$. The reader is referred to textbooks on electronic devices, e.g., [25], for detailed interpretation of these parameters.

```
Rf  b   c   470e3
Ri  b   0   100e3
Re  e   0   22
Rl  ot  0   100e3
Rc  vcc c   10e3
Q1  c   b   e   ebmol
Ci  in  b   0.047e-6
Cf  b   c   250e-12
Co  c   ot  0.47e-6
Vin in  0   1
Vcc vcc 0   9
.model ebmol npn(IS=6.734e-15 BF=200 BR=0.1)
.out V(ot)
```

Figure 4. DK-method netlist for BJT amplifier Fig. 2

B. Common Cathode triode amplifier

A common building block at the front end of many guitar amplifiers, the common cathode triode preamplifier is shown in Figs. 5 and (7). This circuit also provides about 30-dB amplification and saturates with a deep crisp sound compared to the BJT preamp when overdriven for guitar distortion purposes.

The model chosen here is the Koren model [26] with a modification to add grid current.

$$I_g(V_{gk}, V_{pk}) = \log\left(1 + \exp\left(\frac{V_{gk}}{V_T}\right)\right)\frac{V_T}{R_G} \quad (13)$$

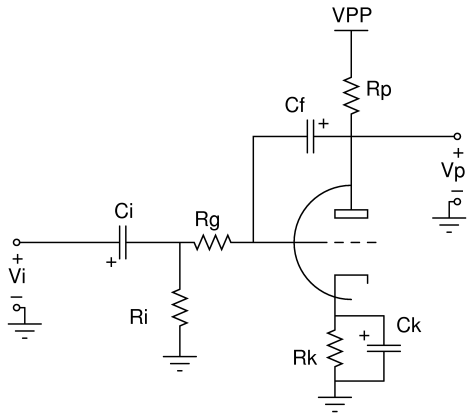

Figure 5. Schematic of the common-cathode amplifier.

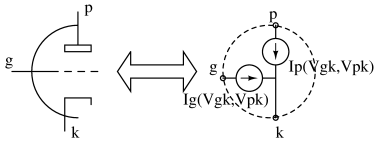

Figure 6. Generic tube device model.

$$I_p(V_{gk}, V_{pk}) = E_d^{K_X} / K_{G1} \left(1 + \mathrm{sign}(E_d)\right), \text{ where} \quad (14)$$

$$E_d = \frac{V_{pk}}{K_P} \log\left(1 + \exp\left(K_P\left(\frac{1}{\mu} + \frac{V_{gk}}{\sqrt{K_{VB} + V_{pk}^2}}\right)\right)\right)$$

The grid current function is chosen to approximate the characteristic of a diode in series with a resistor. Model parameters used are $\mu = 100$, $K_X = 1.4$, $K_{G1} = 1060$, $K_P = 600$, $K_{VB} = 300$, $V_T = 0.026\,\mathrm{V}$, $R_G = 2000\,\Omega$.

C. Solved system nonlinearity

The nonlinearities (7) representing the collector current for the BJT and the plate current for the tube preamp are computed offline and tabulated as shown in Figs. 8, 9. Note that these are two-dimensional functions because in both circuits, the number of controlling voltages is two.

```
Ci   i1   i2      0.047e-6
Ri   i2   0       1e6
Rg   i2   g       70e3
Cf   p    g       2.5e-12
Rp   pp   p       100e3
T1   p    g   k   12ax7
Rk   k    0       1.5e3
Ck   k    0       25e-6
Vi   i1   0       1
Vpp  pp   0       250
.model 12ax7 triode ()
.out V(p)
```

Figure 7. DK-method netlist for tube preamp Fig. 5

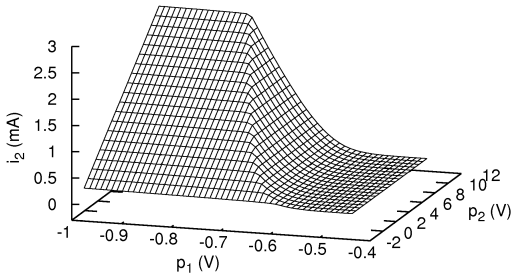

Figure 8. System nonlinearity of BJT preamp

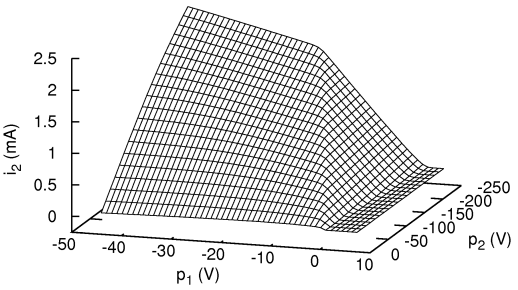

Figure 9. System nonlinearity of tube preamp

V. Experimental Results

In this section we validate correct implementation of the methods described in Part I, and consider the consequences of having a poor approximation for function (7).

A. Sinusoidal test

To validate the correctness of the implementation and to explore the spectral response, the methods were subjected to a 220 Hz sinusoidal excitation. The amplitudes are given in the figure captions. Because the outputs tend to be visually similar when plotted, the error relative to the LTspice implementation is shown on a voltage scale to highlight the differences.

Figures 10 and 11 show the time-domain responses and errors for the BJT preamp. Figures 12 and 13 show the same plots for the tube preamp. The DK methods and the 100x100 table implementations are comparable to LTspice. The coarse tables for both the BJT and tube preamps show significant deviation from the LTspice implementation and are clearly deficient. The time-domain error is sensitive to phase alignment, therefore we turn to a spectral comparison to determine if the errors seen in the time domain translate to audible differences.

The spectral responses to two-tone (110+165 Hz) excitations are shown in Figs. 14 and 15 for the BJT and tube preamps to expose intermodulation effects. Spectral peaks at multiples of the fundamental frequency 55 Hz are connected by straight

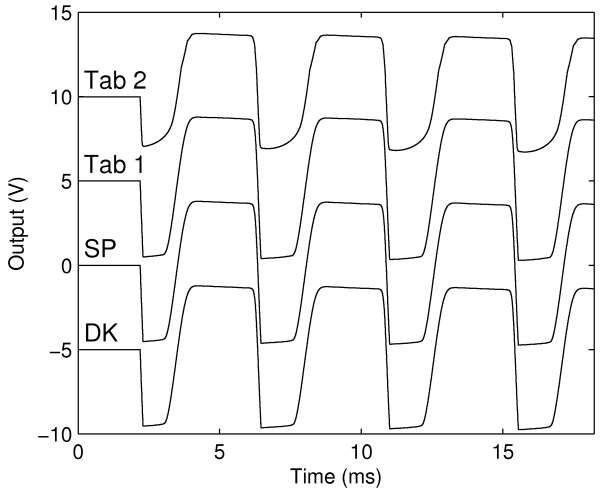

Figure 10. Time-domain response for BJT preamp to 220 Hz, 500 mV excitation. SP is LTspice output plotted without offset, DK is the offline DK-method offset by -5 V, Tab 1 is the DK-method with 100x100 table offset by +5 V, Tab 2 is the DK-method with 50x50 table, offset by +10 V.

lines for visual clarity. Note the rough envelope of the tube preamp compared to the smooth envelope of the BJT preamp. Again the DK-method is comparable to LTspice in both cases. The 100x100 table is nearly identical to LTspice for the BJT. The 100x100 table for the tube preamp shows some errors in the upper frequencies, which may be acceptable for distortion effect purposes.

B. Frequency sweep

The algorithms were subjected to a 20 Hz - 20 kHz sinusoidal exponential frequency sweep. The spectrograms of the outputs give an overview of how well the algorithm performs, illuminating aliasing issues and the sonic quality of the errors produced.

The DK-method is comparable to LTspice for frequencies up to 20 kHz. The 100x100 table appears to be an acceptable compromise for use as a digital plugin. The lower resolution tables exhibit unacceptable errors manifest as broadband spectral noise and are omitted here.

C. Power supply input

The advantage of a K-method approach is that the dimension of the nonlinear lookup table scales with the number of control voltage pairs in the circuit. A straightforward ODE solution requires that each voltage input into the memoryless part of the circuit be considered as an input to the global nonlinearity in the algorithm. This includes all capacitors, inductors, power supplies and inputs. If these are assumed to be constant, and certain capacitances determined to be negligible in the audio frequency, then this can potentially be a lower dimensional nonlinearity, e.g., for a memoryless system with constant power supply, a 1-D nonlinearity as commonly used in waveshaping. However, if these capacitances are important, and the power supply is considered an input, then the K-method type approaches will have a lower dimension nonlinear lookup table.

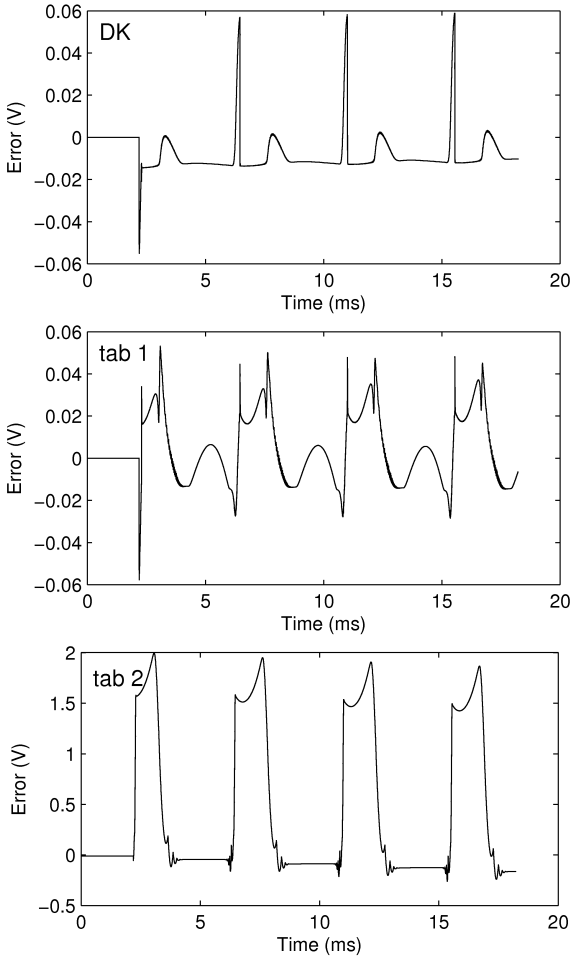

Figure 11. BJT preamp: Time-domain error for 220 Hz, 500 mV excitation, relative to LTspice with default settings for (top to bottom) a) offline DK-method at 8x oversampling, b) DK-method using 100x100 table, c) DK-method using 25x25 table.

Figures 18, 19 demonstrate the ability of the DK-method to incorporate the power supply as an input using the version of the tube preamp algorithm with the 100x100 table. Figure 18 shows the difference between DC power and power supply ripple of 1 V, 60 Hz. The exponential sine sweep in Fig. 19 demonstrates the intermodulation that occurs with a 120 Hz signal on the power supply (a 60 Hz power supply has a fundamental of 120 Hz after rectification). A 5 V excitation is used to exaggerate the effect so that it may be visible above the frequency transform window artifacts. This method may also be used to introduce power supply sag effects due to peaks in the current draw of the power stage.

D. Complexity order analysis

A table-based nonlinearity with uniform sampling has constant time access. Ignoring memory cache issues, with respect to the number of dimensions n, nearest neighbor table lookup scales as $\mathcal{O}(n)$, and linear interpolation scales as $\mathcal{O}(2^n)$.

Accuracy and dynamic range determines the size of table in each dimension. The table-based implementation does have the shortcoming that it scales exponentially with the number of nonlinear dimensions. Assuming C data points per dimension,

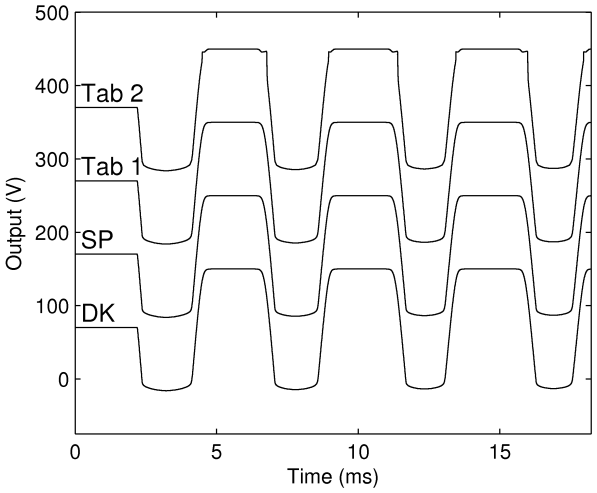

Figure 12. Time-domain response of tube preamp to 220 Hz, 5 V excitation. SP is LTspice output plotted without offset, DK is the offline DK-method offset by -100 V, Tab 1 is the DK-method with 100x100 table offset by +100V, Tab 2 is the DK-method with 50x50 table, offset by +200 V.

storage required scales as $\mathcal{O}(C^n)$. For example with $C = 100$, the highest practical number of dimensions is $n = 4$, where, assuming single precision floating point data, the table size is 400 MB.

Both the BJT and the tube amplifier models have the same dimensions – three states (capacitors), two inputs (signal and power supply), two nonlinear currents, two controlling voltage pairs, and one output. Using (9) and assuming 10 cycles per memory lookup, the cost is approximately 139. Assuming sampling at 384 kHz, and a 1-GHz clock, CPU utilization is 5%. Note that the power supply modulation does not change the computational complexity as the power supply input is a necessary part of the DK-method algorithm.

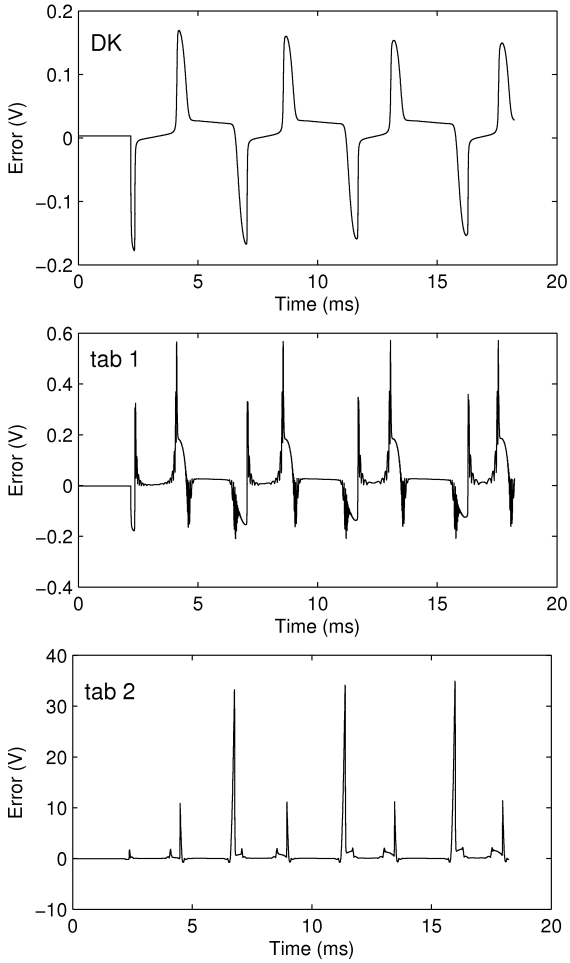

Figure 13. Tube preamp: Time domain error for a 220 Hz, 5 V excitation relative to LTspice with default settings for (Top to bottom) a) offline DK-method at 8x oversampling, b) DK-method using 100x100 table, c) DK-method using 50x50 table.

E. Discussion and potential extensions

The results show that DK-method is essentially equivalent to the LTspice implementation to solve the nonlinear ordinary differential equations of guitar amplifier circuits. The offline DK-method and LTspice both require generally unbounded computational effort to solve the nonlinearities in these circuits, whereas the table versions are capable of running at real-time (50% load on a Core 2 Duo, 2.8 GHz with 8x oversampling to 384 kHz). The prototype code has a much higher load than theoretical because it has been implemented with open-source libraries with no optimization. Similar algorithms have been reported in the literature with a load of 6% at 4x oversampling [15].

A coarse sampling of the system nonlinearity introduces errors that are more prominent at higher frequencies. For the BJT and tube preamps, the 100x100 table seems sufficient as a real-time digital audio effect. The 25x25 table is clearly inadequate. This tradeoff between memory usage and accuracy is a complicated relationship and is typically determined experimentally.

At 8x oversampling, the aliases have fallen off considerably as shown in Figs. 14, 15, which exhibit no alias tones even at -100 dB. In practice, the noise floor of the input signal will also hide low level aliasing.

The tube preamp has wider headroom than the BJT preamp while providing a similar level of gain (around 30 dB). Consequently, the tube amp has a wider linear region than the BJT with sharp cutoffs at the edges. These edges require a fine sampling, while the wide linear region requires a large table range.

If more fidelity is needed, the table resolution must be increased. One could use a finer sampling of the table grid, or use cubic interpolation to improve the function-approximation accuracy. Alternatively, observing the nature of the nonlinearities in Figs. 8 and 9, one could use an adaptive nonuniform table with a finer grid in areas where the rate of change of the table is high, and a sparse grid in regions that are roughly linear. However, a nonuniform table would require a search during lookup, whereas a uniform table allows constant access time. One can also initialize the Newton's method solver using a value from the table, and run a fixed number of additional iterations, which would rapidly increase the accuracy of the result due to the quadratic convergence of the solver. The lookup table should be designed to ensure

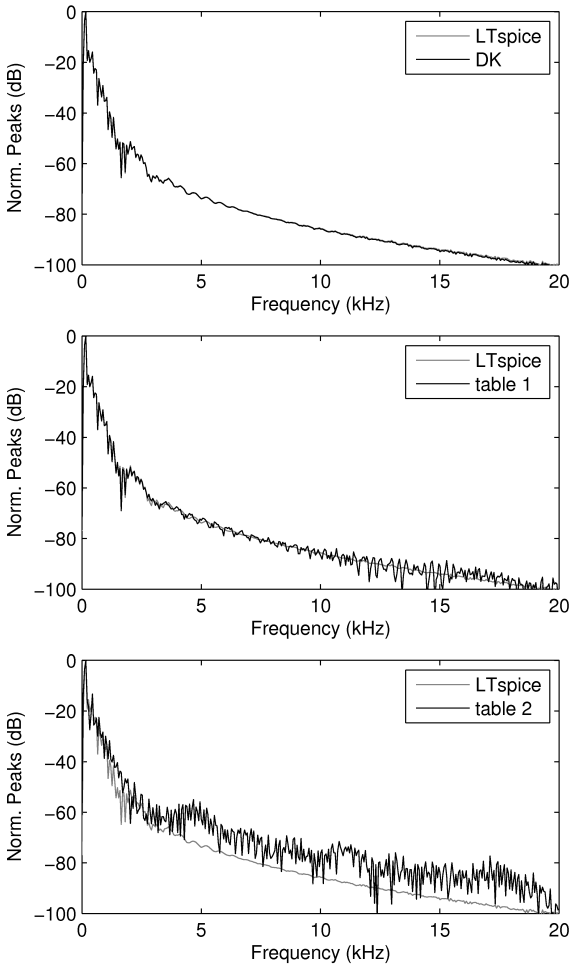

Figure 14. Spectral peaks of the responses to 110+165 Hz, 500 mV excitation for the BJT preamp compared with LTspice result in grey. Top to bottom: DK-method 8x oversampling, 100x100 table, 50x50 table

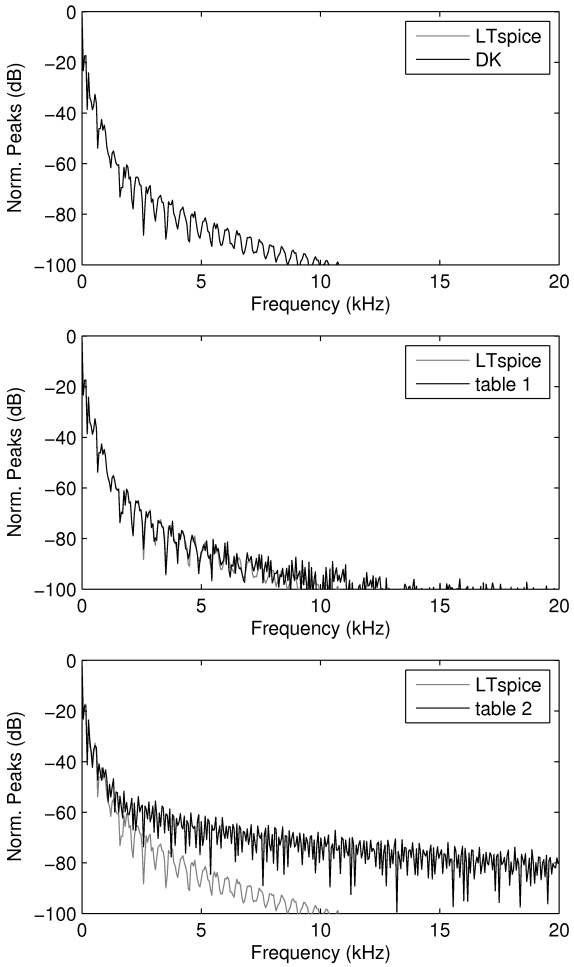

Figure 15. Spectral peaks of the responses to 110+165 Hz, 5 V excitation for the tube preamp, compared with LTspice result in grey. Top to bottom: DK-method 8x oversampling, 100x100 table, 50x50 table

that the iterations start in the Newton method's region of convergence. Finally, the considerable structure of the solved system nonlinearity may also be exploited to use advanced storage-efficient function-approximation methods, for example multidimensional splines, or neural networks, as referenced in Part I. This property suggests that the next major thrust in this research thread should be to develop automated and formula-based approximations to the solved system nonlinearity.

The method as it stands requires large circuits to be divided manually into feedforward blocks with nonlinear dimension 4 or less, which is common practice [15], [24] and acceptable for most circuits given appropriate approximations. Circuits with global feedback can be implemented using a unit delay in the feedback path, which is a good approximation at the 8x oversampling needed to reduce aliasing to an acceptable level in distortion circuits.

Future goals in this research thread should be

1) to detect where coupling exists only in the feedforward direction to decompose approximate subcircuits automatically [21],
2) to improve the function approximation of the global nonlinearity so that there is no restriction on its dimensionality and eliminate the need for blockwise decomposition of the circuit,
3) improved handling of parameter changes to allow real-time parametric circuits (the method was targeted at circuits without time-varying parameters such as guitar amplifiers).

VI. Conclusions

The nodal DK-method is a first step towards automatic generation of real-time filters that simulate the desirable nonlinearities of circuits for musical effects applications and greatly facilitates the process of implementing K-method models of nonlinear systems. The applicability of the method to a BJT preamp circuit and a tube preamp circuit illustrates the generality of the system.

A comparison with equivalent models in SPICE verifies the correctness of our approach and implementation. A suite of tests demonstrates the effects of design choices in the parameters of the methods and illustrate potential artifacts in extreme conditions.

This work demonstrates the efficacy of direct ODE-type methods. compared to WDF or Volterra methods to simulate

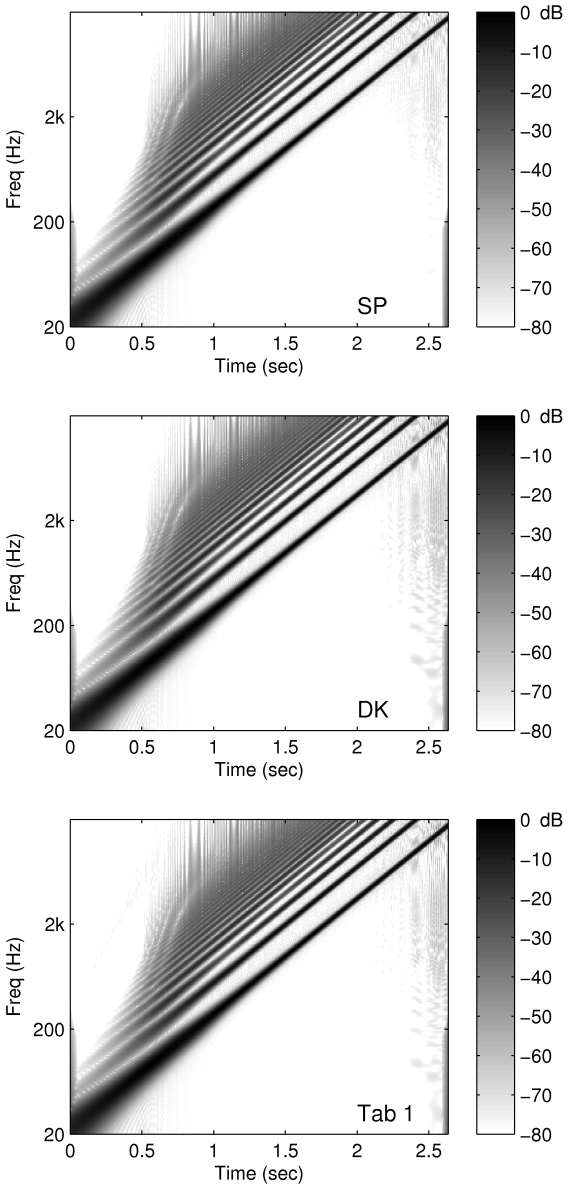

Figure 16. BJT preamp: Spectrograms for exponential frequency sweep from 20 Hz-20 kHz, 500 mV amplitude. LT Spice, DK-method, and 100x100 table.

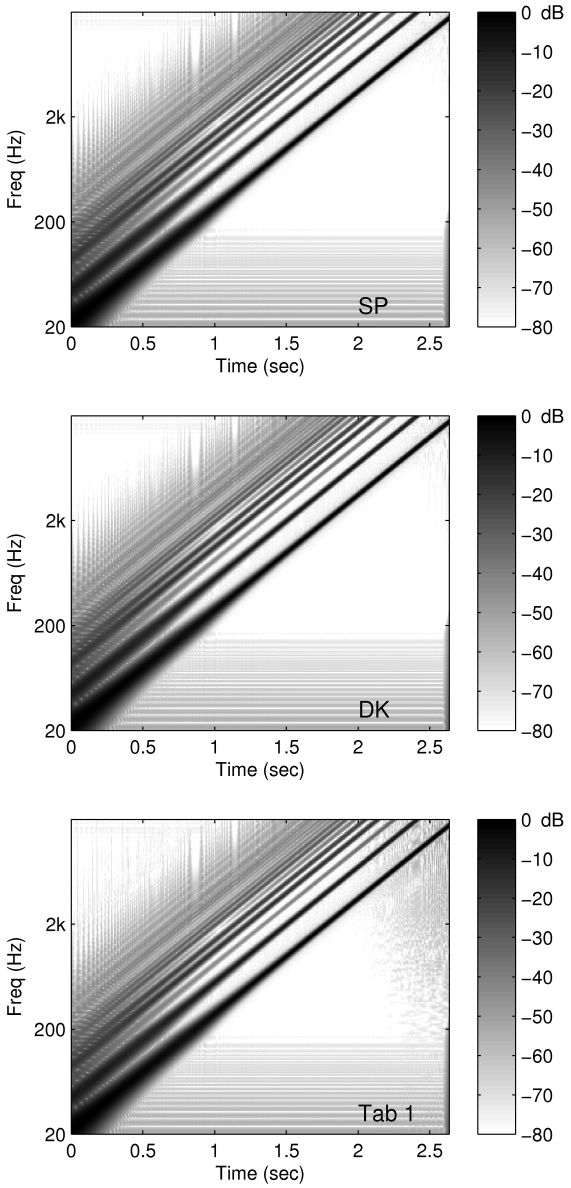

Figure 17. Tube preamp: Spectrograms for exponential frequency sweep from 20 Hz-20 kHz, 5 V amplitude. LT Spice, DK-method, and 100x100 table.

detailed guitar amplifier models in real-time. The K-method family of methods is a special case of ODE-type methods and is similar to direct tabulation of ODE solutions.

If the number of memory elements and inputs is smaller than the number of control voltage pairs in the circuit, then the ODE tabulation approach would yield a lower dimensional table than the K-methods. Otherwise, the K-method family allows opportunities to model the memory of the circuit in greater detail, and introduce secondary inputs, such as power supply hum or sag, without increasing the dimensionality of the system nonlinearity.

In either case, a recursive nonlinear filter in state-space form is the consequence of discretizing the system equations using low-order implicit numerical integration formulas used in stiff ODE solvers. Although the nonlinearities require the use of iterative solvers, these can be computed offline, resulting in a runtime algorithm that is explicit and numerically stable. Iterative solvers are well-known for failing to converge, a trait that is undesirable for real-time effects. The table-based approach provides a deterministic bound on the cost of the algorithm.

Precomputation of the system nonlinearity is essential for real-time effects and this suggests that future researchers should develop storage-efficient methods of approximating this nonlinearity, given its regular structure. Precomputing the system nonlinearity requires that the parameters be fixed, which makes it unsuitable for application to analog synthesizer circuits such as oscillators and time-varying filters. However, parameter changes in the circuit would likely produce nonlinearities that are similar, which can also be incorporated

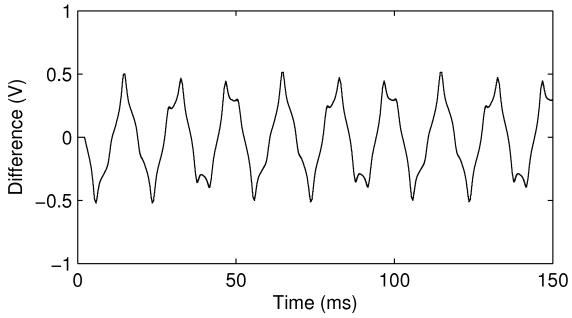

Figure 18. Difference between time-domain response with and without 60 Hz, 1 V ripple on the power supply. DK-method tube preamp, 100x100 table, 220 Hz, 1 V input.

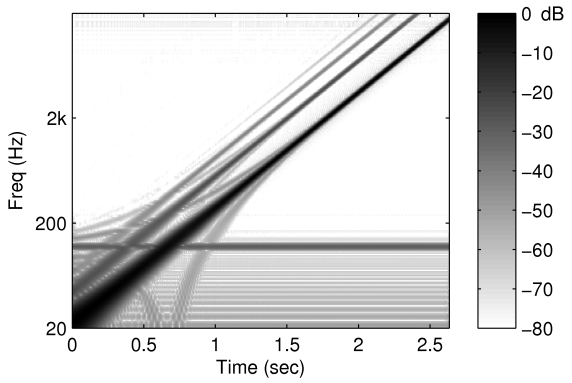

Figure 19. Spectral response of DK-method tube preamp, 100x100 table, to 1 V exponential sine sweep from 20 Hz to 20 kHz, power supply of 250 V with 120 Hz, 5 V ripple to demonstrate modulation.

in the function approximation as an additional dimension. Discovering efficient means to interpolate between parametric sets of system nonlinearities is a potentially rewarding area of interest that could enable time-varying parameters.

REFERENCES

[1] 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," *IEEE Trans. Audio Speech and Lang. Process.*, vol. 18, no. 4, pp. 728–737, May 2010.

[2] J. Pakarinen and D. T. Yeh, "A Review of Digital Techniques for Modeling Vacuum-Tube Guitar Amplifiers," *Computer Music Journal*, vol. 33, no. 2, pp. 85–100, 2009.

[3] J. Pakarinen, V. Välimäki, F. Fontana, V. Lazzarini, 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, 2011, Article ID 940784, 15 pages.

[4] J. Pakarinen and M. Karjalainen, "Enhanced Wave Digital Triode Model for Real-Time Tube Amplifier Emulation," *IEEE Trans. Audio Speech and Lang. Process.*, vol. 18, no. 4, pp. 738–746, May 2010.

[5] 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.

[6] G. De Sanctis and A. Sarti, "Virtual Analog Modeling in the Wave-Digital Domain," *IEEE Trans. Audio Speech and Language Process.*, vol. 18, no. 4, pp. 715–727, May 2010.

[7] J. Macak and J. Schimmel, "Nonlinear Circuit Simulation Using Time-Variant Filter," in *Proc. 12th Int. Conf. on Digital Audio Effects (DAFx-09)*, Como, Italy, Sep. 2009.

[8] J. Pakarinen, M. Tikander, and M. Karjalainen, "Wave Digital Modeling of the Output Chain of a Vacuum-Tube Amplifer," in *Proc. 12th Int. Conf. on Digital Audio Effects (DAFx-09)*, Como, Italy, Sep. 2009.

[9] M. Karjalainen and J. Pakarinen, "Wave digital simulation of a vacuum-tube amplifier," in *IEEE ICASSP 2006 Proc.*, vol. 5, Toulouse, France, 2006, pp. 153–156.

[10] D. T. Yeh and J. O. Smith, "Simulating guitar distortion circuits using wave digital and nonlinear state-space formulations," in *Proc. of the Int. Conf. on Digital Audio Effects (DAFx-08)*, Espoo, Finland, Sept. 1–4, 2008, pp. 19–26.

[11] 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, 2011, Article ID 347645, 15 pages.

[12] A. Fettweis, "Wave digital filters: Theory and practice," *Proc. IEEE*, vol. 74, no. 2, pp. 270 – 327, Feb. 1986.

[13] I. Cohen and T. Hélie, "Real-time simulation of a guitar power amplifier," in *Proc. of the Int. Conf. on Digital Audio Effects (DAFx-10)*, Graz, Austria, Sept. 6–10, 2010.

[14] K. Dempwolf, M. Holters, and U. Zölzer, "Discretization of parametric analog circuits for real-time simulations," in *Proc. of the Int. Conf. on Digital Audio Effects (DAFx-10)*, Graz, Austria, Sept. 6–10, 2010.

[15] J. Macak and J. Schimmel, "Real-Time Guitar Preamp Simulation Using Modified Blockwise Method and Approximations," *EURASIP Journal on Advances in Signal Processing*, vol. 2011, 2011, Article ID 629309, 11 pages.

[16] D. Arfib, "Digital Synthesis of Complex Spectra by Means of Multiplication of Nonlinear Distorted Sine Waves," *Journal of the AES*, vol. 27, no. 10, pp. 757–768, 1979.

[17] S. Möller, M. Gromowski, and U. Zölzer, "A measurement technique for highly nonlinear transfer functions," in *Proc. of the Int. Conf. on Digital Audio Effects (DAFx-02)*, Hamburg, Germany, Sep. 26-28 2002, pp. 203–206.

[18] G. Borin, G. De Poli, and D. Rocchesso, "Elimination of delay-free loops in discrete-time models of nonlinear acoustic systems," *IEEE Trans. Speech and Audio Proc.*, vol. 8, no. 5, pp. 597–605, Sep. 2000.

[19] F. Fontana, F. Avanzini, and D. Rocchesso, "Computation of nonlinear filter networks containing delay-free paths," in *Proc. Conf. on Digital Audio Effects (DAFx-04)*, Naples, Italy, Oct. 2004, pp. 113–118.

[20] F. Fontana and F. Avanzini, "Computation of delay-free nonlinear digital filter networks: Application to chaotic circuits and intracellular signal transduction," *IEEE Trans. Signal Process.*, vol. 56, no. 10, pp. 4703–4715, Oct. 2008.

[21] J. K. White and A. L. Sangiovanni-Vincentelli, *Relaxation Techniques for the Simulation of VLSI CIrcuits*. Boston, MA: Kluwer, 1987.

[22] W. J. McCalla, *Fundamentals of Computer-Aided Circuit Simulation*. Boston: Kluwer Academic Publishers, 1987.

[23] J. W. Eaton, *GNU Octave Manual*. Network Theory Limited, 2002. [Online]. Available: http://www.octave.org/

[24] D. T. Yeh, J. Abel, and J. O. Smith, "Simplified, physically-informed models of distortion and overdrive guitar effects pedals," in *Proc. of the Int. Conf. on Digital Audio Effects (DAFx-07)*, Bordeaux, France, Sept. 10–14, 2007, pp. 189–196.

[25] R. S. Muller, T. L. Kamins, and M. Chan, *Device Electronics for Integrated Circuits*, 3rd ed. Hoboken, NJ: Wiley, 2002.

[26] N. Koren, "Improved vacuum tube models for SPICE simulations," *Glass Audio*, vol. 8, no. 5, p. 18, 1996.

PLACE PHOTO HERE

David T. Yeh David T. Yeh received the B.S. in Electrical Engineering and Computer Sciences at U.C. Berkeley, and the M.S. and Ph.D. degrees in Electrical Engineering from Stanford University in 2009. His thesis topic concerned the modeling of musical distortion circuits such as stompboxes and vacuum tube audio amplifiers primarily used as guitar distortion effects. His research interests include acoustics, music information retrieval, audio signal processing and analog/digital system design. His graduate work had been funded by the Stanford Graduate Fellowship, the National Defense Science and Engineering Graduate Fellowship, and the National Science Foundation Graduate Fellowship.