Research Article

Real-Time Guitar Preamp Simulation Using Modified Blockwise Method and Approximations

Jaromir Macak and Jiri Schimmel

Faculty of Electrical Engineering and Communication, Brno University of Technology, 61600 Brno, Czech Republic

Correspondence should be addressed to Jaromir Macak, jaromir.macak@phd.feec.vutbr.cz

The designing of algorithms for real-time digital simulation of analog effects and amplifiers brings two contradictory requirements: accuracy versus computational efficiency. In this paper, the simulation of a typical guitar tube preamp using an approximation of the solution of differential equations is discussed with regard to accuracy and computational complexity. The solution of circuit equations is precomputed and stored in N-D tables. The stored values are approximated, and therefore different approximation techniques are investigated as well. The approximated functions are used for output signal computation and also for circuit state update. The designed algorithm is compared to the numerical solution of the given preamp and also to the real preamp.

1. Introduction

The real-time digital simulation of analog guitar effects and amplifiers has always been, unfortunately, a compromise between accuracy and speed of a simulation algorithm. There are many different approaches to the simulation, such as a black box approach, a white box (informed) approach, circuit-based approaches, and so forth [1]. All these algorithms differ in accuracy of the simulation as well as in computational complexity. The circuit-based techniques usually offer the best accuracy because a simulated circuit is exactly described by circuit equations, which are often differential equations. Nevertheless, the computational complexity needed for a solution of these equations is very high [2]. Hence, more efficient algorithms for the solution of the equations must be found. This can often be made by neglecting unimportant factors, for example, tube heating, decomposing into separate blocks [3], and decomposing into a linear and nonlinear part. Subsequently, the separate simplified block can be described by a new set of circuit equations that can be solved using several methods, for example, digital filters and waveshaping [3], Hammerstein model [4]. The triode amplifier simulation using wave digital filters is described in [5, 6] and an enhanced tube model

with a grid current is presented in [7]. A decomposition into a linear dynamic and a nonlinear static part is used in nonlinear state-space formulations [8, 9] and this method was automated in [10], where the algorithm parameters are derived from the SPICE netlist. An extended state-space representation was used in [11] for guitar poweramp simulation. A direct solution of nonlinear circuit ordinary differential equations (ODEs) was used in [2, 12], but the computational requirements are high. Therefore, an approximation of the solution of ODEs was used in [13]. However, an approximation of nonlinear functions are also used in [6, 10].

In this paper, a typical guitar preamp is simulated using the approximation of a solution of differential equations. First of all, the guitar preamp is decomposed into separate blocks. It is necessary to find a proper division, because it can influence the accuracy of the whole simulation. A commonly used division into blocks does not consider mutual interactions between connected blocks. As was shown in [13, 14], the simulation fails especially if the load of the simulated block is nonlinear, which is typical for circuits with tubes. Therefore, a modification designed in [13] and investigated in [14] must be used in order to get good simulation results if the blocks are highly nonlinear and connected in series. After

the decomposition, the new set of differential equations is approximated according to [13]. The chosen approximation markedly affects the accuracy of the algorithm as well as the computational complexity. Therefore, several types of approximations will be discussed.

Considering real-time simulation, the computational complexity must be investigated. Commonly used electronic circuit simulators has a computational complexity given by $O(N^{1.4})$ where N is the number of circuit nodes [1]. However, this notation does not determine the real computational complexity, that is, the real number of mathematical operations of the algorithm. Thus in this paper, the whole guitar preamp is numerically simulated using a similar approach as in the electronic circuit simulators and the number of instructions required for one signal sample is computed. It is compared to the number of instructions of the numerical simulation based on the division into blocks and also to the number of instructions of the approximated solution of the given circuit.

Finally, the accuracy of the numerical simulation of the whole preamp must be compared to the simulation using approximation and also to a real guitar preamp.

2. Guitar Preamp Simulation

A guitar tube preamp usually consists of several triode amplifiers connected in series. The separate triode amplifiers are decoupled using decoupling capacitors in order to remove DC. The number of triode amplifiers in the circuit depends on a given topology and therefore it can differ for different preamps. Guitar preamps usually contain two to five triode amplifiers. They are usually connected as a common-cathode amplifier. However, the preamp can also contain a cathode follower (e.g., Marshall). The circuit schematic of a testable guitar tube preamp is shown in Figure 1 and the values for circuit elements are listed in Table 1. The values were chosen as typical values for tube circuits. This preamp consists of four common-cathode triode amplifiers. This circuit schematic is similar to circuit schematics that are used in some real "Hi Gain" guitar amplifiers (e.g., Mesa Boogie or Engl). Only the values of the circuit elements can have differing values, some of the common-cathode amplifiers can have a cathode capacitor to gain the high frequencies and a potentiometer "Gain" with a capacitor connected between terminals of the potentiometer are used instead of resistors R_2 and R_{g2} (see Figure 1). However, the frequency dependence of the "Gain" parameter caused by the capacitor can be simulated by a linear filter.

In order to simulate the preamp from Figure 1, an analysis of the circuit must be done. The circuit equations can be obtained using nodal analysis. In this case, Kirchhoffi Current Law (KCL) is used. After obtaining circuit equations, a discretization of differential equations is realized. The backward Euler method with step of a sampling period was used. Naturally, a different method for discretization can be used, however, this method was chosen due to its simplicity. More information about obtaining the equations can be

Table 1: Values for circuit elements for Figure 1.

R_1	R_{g1}	R_{k1}	R_{p1}	R_2	R_{g2}
68 kΩ	1 MΩ	2.7 kΩ	100 kΩ	470 kΩ	1 MΩ
R_{k2}	R_{p2}	R_3	R_{g3}	R_{k3}	R_{p3}
1.8 kΩ	100 kΩ	470 kΩ	470 kΩ	1.8 kΩ	100 kΩ
R_4	R_{g4}	R_{k4}	R_{p4}	R_L	—
470 kΩ	470 kΩ	1.8 kΩ	100 kΩ	4 MΩ	—
C_1	C_2	C_3	C_4	V_{ss}	—
1 μF	22 nF	22 nF	22 nF	400 V	—

found in [13]. The final circuit equations describing the circuit in Figure 1 are

$$0 = \left(V_{in} - V_{g1}\right)G_1 - V_{g1}G_{g1} - i_{g1},$$

$$0 = V_{c1m} - V_{c1} - \frac{V_k G_k - i_{p1} - i_{g1}}{C_1 f_s},$$

$$0 = \left(V_{ss} - V_{p1}\right)G_{p1} - \left(V_2 - V_{g2}\right)G_2 - i_{p1},$$

$$0 = V_{c2m} - V_{c2} - \frac{\left(V_2 - V_{g2}\right)G_2}{C_2 f_s},$$

$$0 = \left(V_2 - V_{g2}\right)G_2 - V_{g2}G_{g2} - i_{g2},$$

$$0 = V_{k2}G_{k2} - i_{g2} - i_{p2},$$

$$0 = \left(V_{ss} - V_{p2}\right)G_{p2} - \left(V_3 - V_{g3}\right)G_3 - i_{p2},$$

$$0 = V_{c3m} - V_{c3} - \frac{\left(V_3 - V_{g3}\right)G_3}{C_3 f_s}, \quad (1)$$

$$0 = \left(V_3 - V_{g3}\right)G_3 - V_{g3}G_{g3} - i_{g3},$$

$$0 = V_{k3}G_{k3} - i_{g3} - i_{p3},$$

$$0 = \left(V_{ss} - V_{p3}\right)G_{p3} - \left(V_4 - V_{g4}\right)G_4 - i_{p3},$$

$$0 = V_{c4m} - V_{c4} - \frac{\left(V_4 - V_{g4}\right)G_4}{C_4 f_s},$$

$$0 = \left(V_4 - V_{g4}\right)G_4 - V_{g4}G_{g4} - i_{g4},$$

$$0 = V_{k4}G_{k4} - i_{g4} - i_{p4},$$

$$0 = \left(V_{ss} - V_{p4}\right)G_{p4} - V_{p4}G_L - i_{p4},$$

where i_g is the grid current function $i_g(U_g - U_k)$ and i_a is the plate current function $i_a(U_g - U_k, U_a - U_k)$. The symbol G represents the conductance of the resistors from the circuit schematic in Figure 1, f_s is the sampling frequency, V_{ss} is the power supply, and V_{in} is the input voltage. Voltages V_{c1m}, V_{c2m}, V_{c3m}, and V_{c4m} are the voltages on the capacitors in the previous signal sampling period. Equations (1) have 15 unknown voltage variables (denoted in Figure 1); 1st, 5th, 9th, 13th equations describe grid nodes of all tubes four
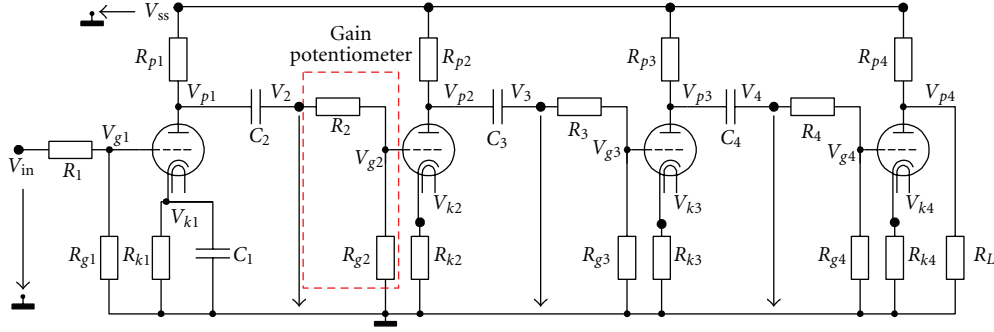
FIGURE 1: A guitar tube preamp circuit schematic.

tubes; 2nd, 6th, 10th, and 14th equations are cathode nodes and 3rd, 7th, 11th, 15th equations are anode nodes. Koren's nonlinear model of a triode has been used in this paper [15]. The triode plate current is given by

$$i_a = \frac{E_1^{E_x}}{K_{g1}}\left(1 + \text{sgn}(E_1)\right), \tag{2}$$

where

$$E_1 = \frac{U_{ak}}{K_p}\log\left(1 + \exp\left(K_p\left(\frac{1}{\mu} + \frac{U_{gk}}{\sqrt{K_{vb} + U_{ak}^2}}\right)\right)\right). \tag{3}$$

Parameters μ, E_x, K_{g1}, K_{g2}, K_p, and K_{vb}, are available in [15], U_{gk} is the grid-to-cathode voltage and U_{ak} is the plate-to-cathode voltage. The grid current is not specified in [15], and therefore the grid model was adopted from the Microcap simulator [16]. The grid current is

$$i_g = \begin{cases} g_{cf}\left(u_{gk} - g_{co}\right)^{3/2}, & u_{gk} \geq g_{co}, \\ 0, & u_{gk} < g_{co}, \end{cases} \tag{4}$$

where $g_{cf} = 1 \cdot 10^{-5}$ and $g_{co} = -0.2$. Frequency properties of the tube (e.g., Miller capacitance) are not considered in the simulation due to simplicity of tube model. However, it influences accuracy of the simulation, because the Miller capacitance makes a low-pass filter at the input of the tube. [17]

Generally, a system of nonlinear equation (1) is solved using the Newton-Raphson method

$$\mathbf{x}_n^{i+1} = \mathbf{x}_n^i - \mathbf{J}^{-1}\left(\mathbf{x}_n^i\right)\mathbf{F}\left(\mathbf{x}_n^i\right), \tag{5}$$

where $\mathbf{x}_n^i$ is a vector of unknown voltages, $J(\mathbf{x}_n^i)$ is the Jacobian matrix, and $\mathbf{F}(\mathbf{x}_n^i)$ is the function given by (1), i denotes iteration index and n is the time index.

However, if it is a real-time simulation, computational demand has to be investigated. According to (1), the function $\mathbf{F}(\mathbf{x}_n^i)$ involves four grid functions (4), four plate functions (2), 47 add operations, and 31 multiply operations if the sampling frequency f_s is substituted with the sample period $T_s = 1/f_s$ and values of capacitors are substituted with the reciprocal values. Considering the computational cost

of nonlinear device model functions (2) and (4) as c_p and c_g respectively, the total cost of the function $\mathbf{F}(\mathbf{x}_n^i)$ is $c_f = 78 + 4c_p + 4c_g$ operations. The Jacobian matrix $\mathbf{J}(\mathbf{x}_n^i)$ contains partial first-order derivation of the function $\mathbf{F}(\mathbf{x}_n^i)$. Since the function $\mathbf{F}(\mathbf{x}_n^i)$ has not been a continuous function, the derivations are computed for instance using the finite difference formula

$$f'(x) = \frac{1}{h}\left(f(x + h) - f(x)\right) \tag{6}$$

with a step h that consists of two function calls $\mathbf{F}(\mathbf{x}_n^i)$ 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. 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 a more efficient implementation. Then, (5) is rewritten as

$$\mathbf{LU} = \mathbf{J}\left(\mathbf{x}_n^i\right), \tag{7}$$

$$\mathbf{Ly} = \mathbf{F}\left(\mathbf{x}_n^i\right), \tag{8}$$

$$\mathbf{U}\Delta\mathbf{x}_n^i = \mathbf{y}, \tag{9}$$

$$\mathbf{x}_n^{i+1} = \mathbf{x}_n^i - \Delta\mathbf{x}_n^i. \tag{10}$$

According to [18], the cost of (7) (Crout's algorithm) is $(1/3)N^3$ of inner loops containing one multiply and add operation, the cost of (8) and (9) is N^2 multiply and add operations. Thus, the total cost is $c_{LU} = (2/3)N^3 + 4N^2$ operations. Knowing $\Delta\mathbf{x}_n^i$, (10) can be solved, which requires N add operations. The total cost of the Newton method is then

$$c_{nm} = i\left(\left(c_f + 3\right)N + c_f + \frac{2}{3}N^3 + 4N^2 + N\right), \tag{11}$$

where i is a number of iterations of the Newton method and N is the number of circuit nodes. However, it must be said that this number is theoretical. Neither of the algorithm branches nor memory movements have been considered.
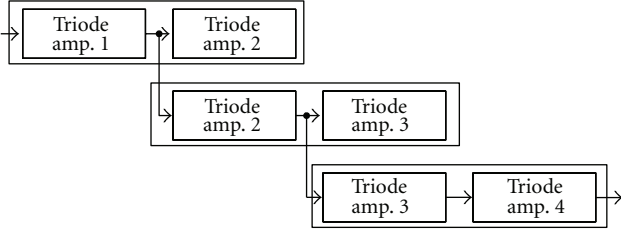
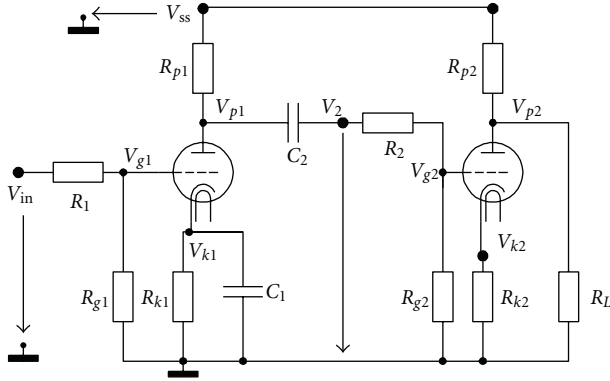FIGURE 2: Preamp block decomposition using modified blockwise method [13].

FIGURE 3: Circuit schematic of the first block of the guitar preamp.

3. Simulation Using Modified Blockwise Method

The simulated circuit can also be divided into blocks. An example of division into blocks using the modified blockwise method [14] is shown in Figure 2. The whole circuit from Figure 1 is divided into three separate blocks containing two tubes in this case. Two types of blocks are used. The first one contains the common-cathode tube amplifier with a cathode capacitor and is used as the first block in Figure 2. The other one contains the common-cathode tube amplifier without the cathode capacitor and is used as the second and third block. Then, the simulation requires a solution of three independent simpler blocks, and as a result the computational complexity can be lower.

The circuit schematic of the first block is shown in Figure 3. It consists of two common-cathode tube amplifiers, the first tube amplifier is connected exactly according to the circuit schematic in Figure 1. The second tube amplifier is similar to the second tube amplifier in 1, only the second decoupling capacitor is not included and the load resistor R_L is connected directly to plate of the second tube. This can be done, because the value of resistor R_L and presence of decoupling capacitor have very low influence on the first preamp [14] because the output signal is obtained at the output of the first amplifier and the second amplifier builds only the load of the first amplifier. The value of resistor R_L is 4 MΩ and the others circuit elements values can be obtained from Table 1. The output signal voltage is obtained from the node V_2.

The circuit is described by

$$0 = \left(V_{\text{in}} - V_{g1}\right)G_1 - V_{g1}G_{g1} - i_{g1},$$

$$0 = V_{c1m} - V_{c1} - \frac{V_k G_k - i_{p1} - i_{g1}}{C_1 f_s},$$

$$0 = \left(V_{\text{ss}} - V_{p1}\right)G_{p1} - \left(V_2 - V_{g2}\right)G_2 - i_{p1},$$

$$0 = V_{c2m} - V_{c2} - \frac{\left(V_2 - V_{g2}\right)G_2}{C_2 f_s}, \quad (12)$$

$$0 = \left(V_2 - V_{g2}\right)G_2 - V_{g2}G_{g2} - i_{g2},$$

$$0 = V_{k2}G_{k2} - i_{g2} - i_{p2},$$

$$0 = \left(V_{\text{ss}} - V_{p2}\right)G_{p2} - V_{p2}G_L - i_{p2}.$$

The function $\mathbf{F}(\mathbf{x}_n)$ 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, (11) can be used for the determination of the computational complexity. For given $N = 7$, the total cost is

$$c_{nm} = i\left(749 + 16\left(c_g + c_p\right)\right). \quad (13)$$

Figure 4 shows the circuit schematic of the second and third block. The circuit elements values can be derived from Table 1. The block contains two common-cathode tube amplifiers without the cathode capacitor. The output signal is obtained from the node V_2.

The computational complexity is similar to the first block. They only differ in the function $\mathbf{F}(\mathbf{x}_n)$, because the cathode capacitor in the first amplifier is missing (circuit equations are not explicitly shown). Therefore, it has $c_f = 34 + 2c_p + 2c_g$ operations in this case. The cost is then

$$c_{nm} = i\left(728 + 16\left(c_g + c_p\right)\right). \quad (14)$$

If the blocks are connected together, the total cost of the solution of these blocks is

$$c_{nm} = i_1\left(749 + 16\left(c_g + c_p\right)\right) + \\ + (i_2 + i_3)\left(728 + 16\left(c_g + c_p\right)\right), \quad (15)$$

where i_1, i_2, and i_3 are numbers of iterations of the first, second, and third block, respectively.

4. Simulation Using Modified Blockwise Method and Approximation

A system of differential equations can be described by the equation

$$\mathbf{0} = \mathbf{F}(V_{\text{in}}, V_{c1}, V_{c2}, \dots, V_{cM}), \quad (16)$$

where V_{in} is an input voltage or an input signal value and $V_{c1}, V_{c2}, \dots, V_{cM}$ are voltages on capacitors. Thus, the system
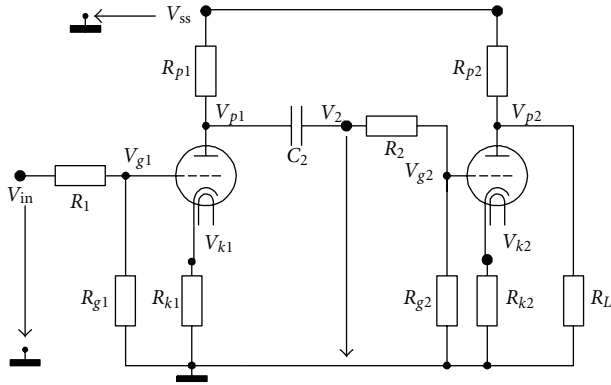
FIGURE 4: Circuit schematic of the second block of the guitar preamp.

has $N = M + 1$ inputs and also N outputs—an output voltage or an output signal level and new voltages on the capacitors. The system has a particular solution for a combination of the inputs. The solution can be precomputed for certain combinations of the input voltages and it must be approximated, because the solution has to involve all combinations of the input voltages [13]. The output signal value is computed on the basis of the approximated solution. Then, the state of the circuit (capacitor voltages) is actualized. The new capacitor voltages are used as the inputs for the next sample period. The approximation leads to the following set of equations:

$$
\begin{aligned}
V_{\text{out}}^n &= F_{\text{out}}\left(V_{\text{in}}^n, V_{c1}^n, V_{c2}^n, \ldots, V_{cM}^n\right), \\
V_{uc1}^{n+1} &= V_{uc1}^n + T_s F_{c1}\left(V_{\text{in}}^n, V_{c1}^n, V_{c2}^n, \ldots, V_{cM}^n\right), \\
V_{uc2}^{n+1} &= V_{uc2}^n + T_s F_{c2}\left(V_{\text{in}}^n, V_{c1}^n, V_{c2}^n, \ldots, V_{cM}^n\right), \\
&\vdots \\
V_{ucM}^{n+1} &= V_{ucM}^n + T_s F_{cN}\left(V_{\text{in}}^n, V_{c1}^n, V_{c2}^n, \ldots, V_{cM}^n\right),
\end{aligned}
\tag{17}
$$

where functions $F_{\text{out}}, F_{uc1}, F_{uc2}, \ldots, F_{ucM}$ are approximating functions, T_s is the sample period, and the superscript n denotes time index. The function F_{out} approximates directly the output signal value and the functions $F_{uc1}, F_{uc2}, \ldots, F_{ucM}$ approximate changes of the capacitor voltage. The computational complexity depends on the number of accumulation circuit elements (capacitors) N. 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. Furthermore, the chosen approximation influences the accuracy of the algorithm and also memory requirements.

(i) *Linear Interpolation* offers the fastest implementation. However, it requires many precomputed values for smooth behavior. The linear interpolation is given by

$$
V_{\text{out}} = V_{\text{out}}^i (1 - p) + V_{\text{out}}^{i+1} p,
\tag{18}
$$

where V_{out}^i are precomputed output values, i is index into a vector of fprecomputed output values and p is a fractional part between neighbouring precomputed values [19]. The linear interpolation consists of two add operations, two multiply operations, index and fractional part computation. The bilinear interpolation has to be used as the approximating function for the first-order differential system, because the system has two inputs: input signal and the capacitor voltage. The bilinear interpolation requires three linear interpolations (18). Generally, the N-dimensional linear interpolation must be used, if there are M accumulation elements. It requires $2^N - 1$ linear interpolations.

(ii) *Spline Interpolation* offers the smooth behavior of an approximating function. The spline interpolation is given by

$$
\begin{aligned}
V_{\text{out}} = {}& \frac{p^3}{6} V_{\text{out}}^{i-1} + \frac{(1+p)^3 - 4p^3}{6} V_{\text{out}}^i \\
& + \frac{(2-p)^3 - 4(1-p)^3}{6} V_{\text{out}}^{i+1} + \frac{(1-p)^3}{6} V_{\text{out}}^{i+2},
\end{aligned}
\tag{19}
$$

where p is the fractional part between neighbouring precomputed values [19]. It consists of 10 add operations and 19 multiply operations, index and fractional part computation. The N-dimensional spline interpolation requires $(4^N - 1)/3$ spline interpolations (19).

(iii) *Cubic Spline Approximation* also offers the smooth behavior of approximating function but compared to the spline interpolation, coefficients of cubic polynomials are stored instead of the precomputed values. The cubic polynomial is given by

$$
\begin{aligned}
V_{\text{out}}^i &= a_j V_{\text{in}}^3 + b_j V_{\text{in}}^2 + c_j V_{\text{in}} + d_j \\
&= \left(\left(a_j V_{\text{in}} + b_j\right) V_{\text{in}} + c_j\right) V_{\text{in}} + d_j,
\end{aligned}
\tag{20}
$$

where j index is into a vector of polynomial coefficients [20]. It consists of three add operations and three multiply operations. The combination of spline and linear approximation can be used for the higher-order system simulation, because the functions that are being approximated have very similar shape with different capacitor voltages $V_{c1}, V_{c2}, \ldots, V_{cN}$, as shown in simulations in [13]. The splines are used for different input signal values and the linear interpolation (18) for different capacitor voltages. Two spline approximations (20) and one linear interpolation (18) are required for the first-order system ($N = 2$). Generally, it requires $2^{(N-1)} - 1$ linear interpolations and $2^{(N-1)}$ spline approximations.

All approximations and interpolations work with indexes into a vector of spline coefficients or precomputed values of

the solution of the system. It is necessary to find efficient determination of the index, if it is real-time simulation. The fastest way of determination of the index is computation directly from the input values. The solution of the system is precomputed for integer values of the inputs. Then the index is obtained from

$$i = \lfloor V_{\text{in}} \rfloor \qquad (21)$$

and the fractional part is obtained from

$$p = V_{\text{in}} - i. \qquad (22)$$

The system of (12) has three inputs (input voltage and two input voltages on capacitors), and therefore $M = 2$ and $N = 3$ and three approximating functions are needed. The solution was precomputed (error limit of Newton method was 0.0001). The approximating functions F_{out}, F_{c1}, and F_{c2} are plotted in Figure 5 for 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. Naturally, the capacitor voltages can have any value between 0 V and the power supply voltage.

The individual functions have very similar shapes with different capacitor voltages. Therefore, splines are used for the approximation with constant capacitor voltages and different input voltage V_{in} and then, computed spline coefficients are stored in look-up table in a row r that was computed as a linear function of the input and capacitor voltages

$$r = V_{\text{in}} V_{c1_{\text{steps}}} V_{c2_{\text{steps}}} + V_{c1} V_{c2_{\text{steps}}} + V_{c2}. \qquad (23)$$

Then, the spline coefficients are computed for different capacitor voltage values. Subsequently, linear interpolation between spline curves is used in order to get the final values. It was experimentally found that the coefficient computation can be made for two values of the capacitor voltage V_{C1} (minimal value 0 V and maximal possible value that depends on values of resistors in the circuit). 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 ± 200 V with a step of 1 V (see Table 2). The input voltages can be of course lower than 1 V, but in this range the approximated function is almost linear, and therefore it can be approximated by the spline with low error. The maximal chosen deviation between numerical solution and approximation was 0.1 V. Index into the table of spline coefficients is dependent on the input voltage and capacitor voltages and a number of rows r of the table can be obtained from

$$n_r = V_{\text{in}_{\text{steps}}} V_{c1_{\text{steps}}} V_{c2_{\text{steps}}} \qquad (24)$$

and in this case, 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 is 6 MB. The final equation for the simulation of this block are

$$V_{\text{out}}^n = F_{\text{out}}(V_{\text{in}}^n, V_{c1}^n, V_{c2}^n),$$
$$V_{uc1}^{n+1} = V_{uc1}^n + T_s F_{c1}(V_{\text{in}}^n, V_{c1}^n, V_{c2}^n), \qquad (25)$$
$$V_{uc2}^{n+1} = V_{uc2}^n + T_s F_{c2}(V_{\text{in}}^n, V_{c1}^n, V_{c2}^n).$$

TABLE 2: Lookup table for simulation of the first block (size 2 MB).

Variable	Min.	Max.	Step
V_{in} [V]	-200	200	1
V_{C1} [V]	0	5	5
V_{C2} [V]	0	400	5

TABLE 3: Look-up table for simulation of the second and third block (size 1 MB).

Variable	Min.	Max.	Step
V_{in} [V]	-200	200	1
V_{C1} [V]	0	400	5

TABLE 4: Computational complexity comparison of simulations based on the Newton method—number of operations.

Simulation type	One iteration	Maximal iteration	Average iteration
Whole	5.48×10^3	5.48×10^5	1.53×10^4
By blocks	2.97×10^3	1.39×10^4	6.89×10^3

The circuit schematic from Figure 4 has only two inputs—input voltage V_{in} and capacitor voltage V_{C1}. Therefore, only two approximating functions (F_{out} and F_{c1}) are needed (see Figure 6). They were approximated using the same technique as the functions in the previous circuit. The input voltage grid was ± 200 V with a step of 1 V and capacitor voltage V_{c1} grid was between 0 V and the power supply voltage with a step of 5 V (see Table 3). The total size of both tables is 2 MB in this case. The final equations for the simulation of this block are

$$V_{\text{out}}^n = F_{\text{out}}(V_{\text{in}}^n, V_{c1}^n),$$
$$V_{uc1}^{n+1} = V_{uc1}^n + T_s F_{c1}(V_{\text{in}}^n, V_{c1}^n). \qquad (26)$$

The final simulation equations (25), (26) are quite simple. This is the biggest advantage when comparing with other methods for real time simulation, for example, the state space method, which requires matrix operations and also nonlinear function precomputation stored N-D lookup table.

5. Computational Complexity

The proposed algorithms were compared with regard to the computational complexity. For this purpose, the functions (2) and (4) of the nonlinear device model were tabulated and interpolated using the linear interpolation. As a result, the cost c_g is two add operation, and two multiply operations, the cost c_p is six add operations and six multiply operations. Table 4 shows a number of operations required for the simulations based on the Newton method. Since the Newton method is an 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 maximal number of iterations depend on the type of the input signal. Therefore,
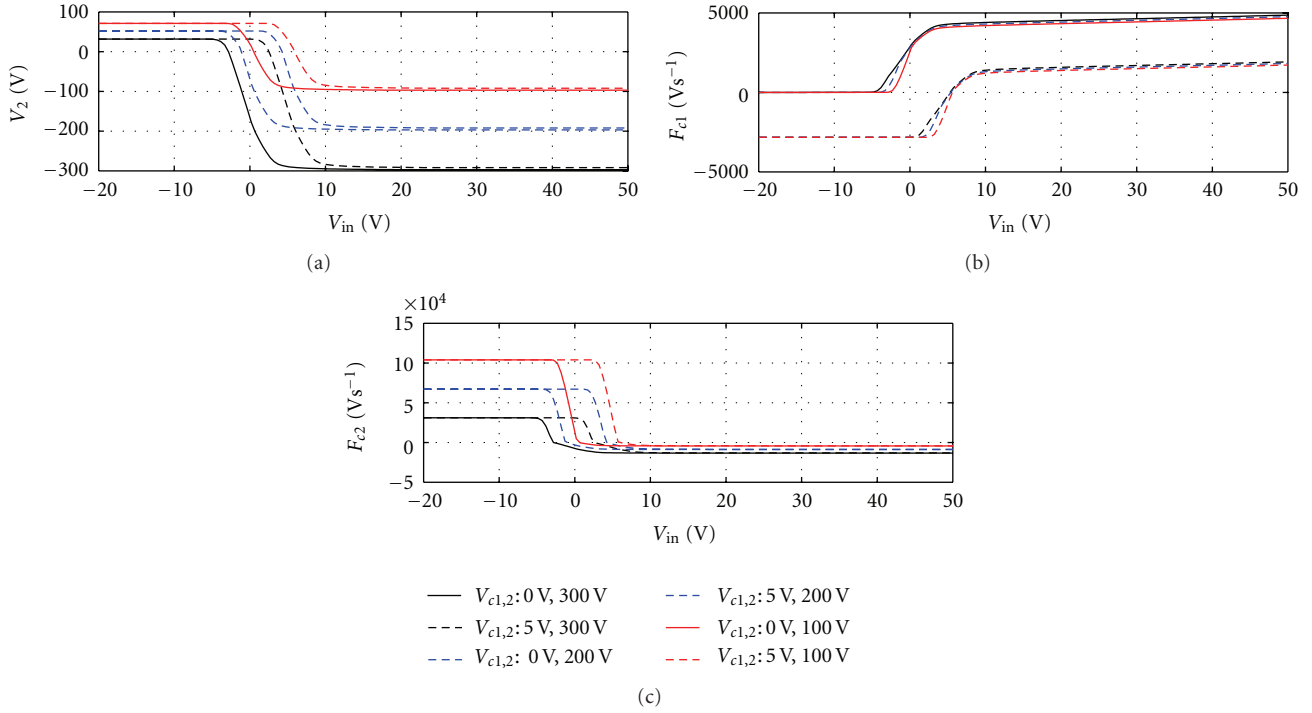
(a)

(b)

(c)

FIGURE 5: Approximating functions for the simulation of the system (12)—output function (a), capacitor C_1 up-date (b) and capacitor C_2 update (c) functions.
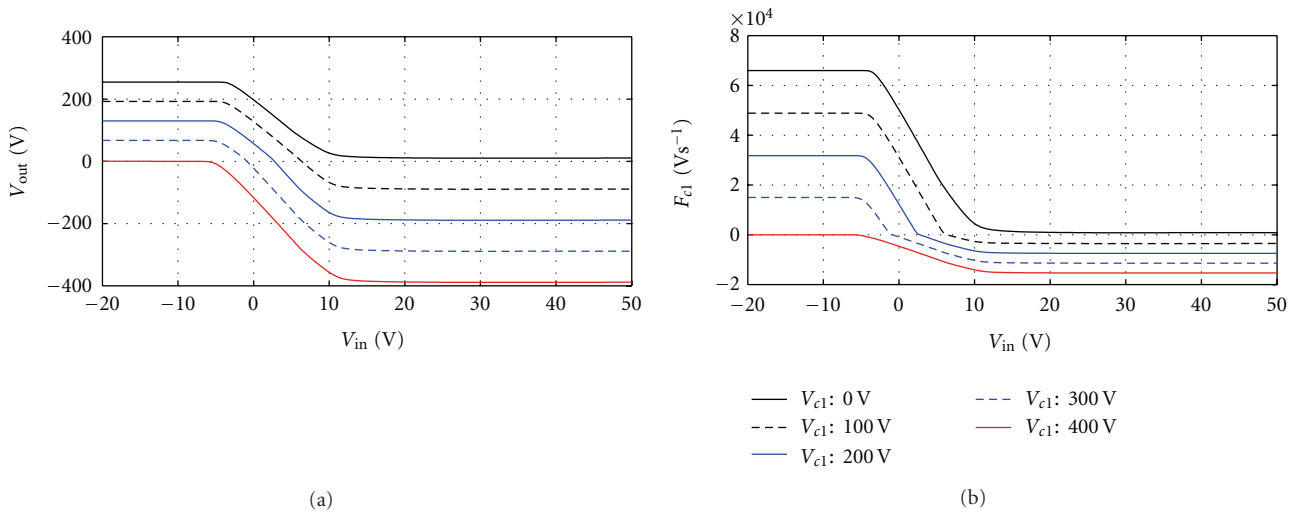

(a)

(b)

FIGURE 6: Approximating functions for the simulation of the system (12)—output function (a) and capacitor C_2 update function (b).

the algorithms were tested with an E-chord guitar riff with maximal amplitude around 200 mV. The whole circuit simulation required 2.79 iterations on average and 100 iterations at the most. In the case of the simulation using block decomposition, the 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 iteration of individual block differs because each block processes a different signal. The maximum number of iteration was 100 and the error limit of

the Newton method was 1×10^{-5}. The number of iteration was computed from the whole signal (5 s, 240×10^3 samples).

The computational complexity of algorithms based on approximations is shown in Table 5. There are available results for the whole preamp simulation as well as for the blockwise simulation. The numbers were computed from (25) and (26) where different types of approximation of appropriate order N from Section 4 were used. Similarly, the whole system can be approximated by order of approximation $N = 5$, because the whole circuit contains 4 capacitors.

However, the whole circuit simulation was not implemented due to complex approximating functions and also the look-up table size would be huge.

As the results available in Tables 4 and 5 have shown, the algorithms based on approximation offer constant computational complexity, which is also much lower than at the algorithms based on the Newton method. The linear interpolation has the lowest computational complexity. However, due to higher memory demands it is not suitable and therefore the spline approximation was chosen as the best method.

6. Simulation Results

The algorithm was implemented in C++ language as the VST plug-in effect and then tested in realtime. It was tested with a 2.66 GHz i7 Intel Mac with 4 GB RAM at a sampling frequency of 48 kHz using an external audio interface M-Audio Fast Track Pro with the ASIO buffer size 128 samples. If no oversampling was used, the CPU load was around 3%. To reduce aliasing distortion, 4-x oversampling was implemented. Polyphase FIR interpolation and decimation was used. The CPU load with 4-x oversampling was around 6%. The proposed algorithm was then tested with different input signals including a sinusoid signal at different frequencies and amplitudes (including frequencies close to the Nyquist frequency), logarithmic sweep signal (see Figure 7) and also a 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. Comparison between the simulation of whole preamp using the Newton method and the simulation based on the spline approximation is plotted in Figure 8. Time difference signals are normalized to the maximal value of the output signal.

The error values, such as maximal and average error, are also expressed in Table 6. The maximal error is around 2 V. An amplification of the preamp is approximately 1.84×10^4 and measured level of noise at the output of the preamp without connected guitar is approximately 4.7 V_{pp} and with connected guitar it is approximately 32 V_{pp} (these high values are caused by extremely high amplification, normally, the amplification is lower). Therefore the error can be masked. 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.

The circuit was also simulated with a reduced power supply voltage to 261 V and the results were compared to a real home-made guitar preamp connected according to the circuit schematic in Figure 1 with the reduced power supply. Firstly, the simulations using Newton method and approximations were compared in Figure 9. The error was approximately the same as in Figure 8—the maximum error increased from 2 V to 3.5 V but the average error decreased from 6.20×10^{-3} to 4.50×10^{-3} V. The relative error in Figure 9 is higher due to lower power supply. The comparison between the output of real circuit and its

TABLE 5: Computational complexity comparison of simulations based on approximations—number of operations.

Simulation type	Linear interpretation	Spline Interpertation	Spline Approx
Whole	9.38×10^2	5.28×10^4	9.53×10^2
By blocks	2.06×10^2	2.58×10^3	2.51×10^2

TABLE 6: Errors for simulation from Figure 8. The plate voltage signal errors are displayed.

—	V_{p1}	V_{p2}
Max [V]	6.27×10^{-4}	2.38×10^{-1}
Mean [V]	1.13×10^{-5}	5.70×10^{-3}
var [V]	1.05×10^{-7}	2.54×10^{-4}
—	V_{p3}	V_{p4}
Max [V]	2.12×10^{-1}	1.99
Mean [V]	1.01×10^{-2}	6.20×10^{-3}
var [V]	1.07×10^{-3}	0.13×10^{-3}

TABLE 7: Harmonics comparison from Figure 11. The magnitudes are related to the first harmonic.

—	2	3	4	5	6
Meas. [dB]	−35.1	−10.0	−36.1	−14.7	−37.7
Sim. [dB]	−27.4	−9.8	−28.3	−14.7	−29.5
Diff. [dB]	7.6	0.2	7.7	0.1	8.2

digital simulation was made using sinusoid signal because a harmonic signal generator was used as a signal source for the guitar preamp. The output signal was recorded using soundcard. The input harmonic signal had an amplitude of 150 mV and a frequency of 1 kHz. Measured and simulated time-domain signals are shown in Figure 10. Most of the errors occur on transients, but the testing preamp was extremely noisy, and therefore it is very hard to determine the deviation of the simulation. The spectrum of the signals is shown in Figure 11. The rectangular window with a length of a hundredfold of the signal period (48 samples at a sampling frequency of 48 kHz) was used to minimize spectrum leakage. The measured preamp was homemade and it was not shielded. Therefore, intermodulation distortion components occur in measured spectrum. Table 7 shows magnitudes of the higher harmonic components related to the first harmonic. The error between measurement and simulation is shown as well. The odd harmonics were almost the same, but the even harmonic were higher in the simulation. However, this deviation can be caused by the tube model, because the same simulation results were obtained using the numerical solution of the whole circuit. The general model of a 12ax7 tube was used in the simulation but the real tubes have different properties—there are many different types of 12ax7 tube and also the same type can differ because of a manufacturer's tolerance. In order to get an accurate simulation, the general tube model must be tuned according to the used tubes. However, it requires a measurement of the transfer functions of the used tubes.
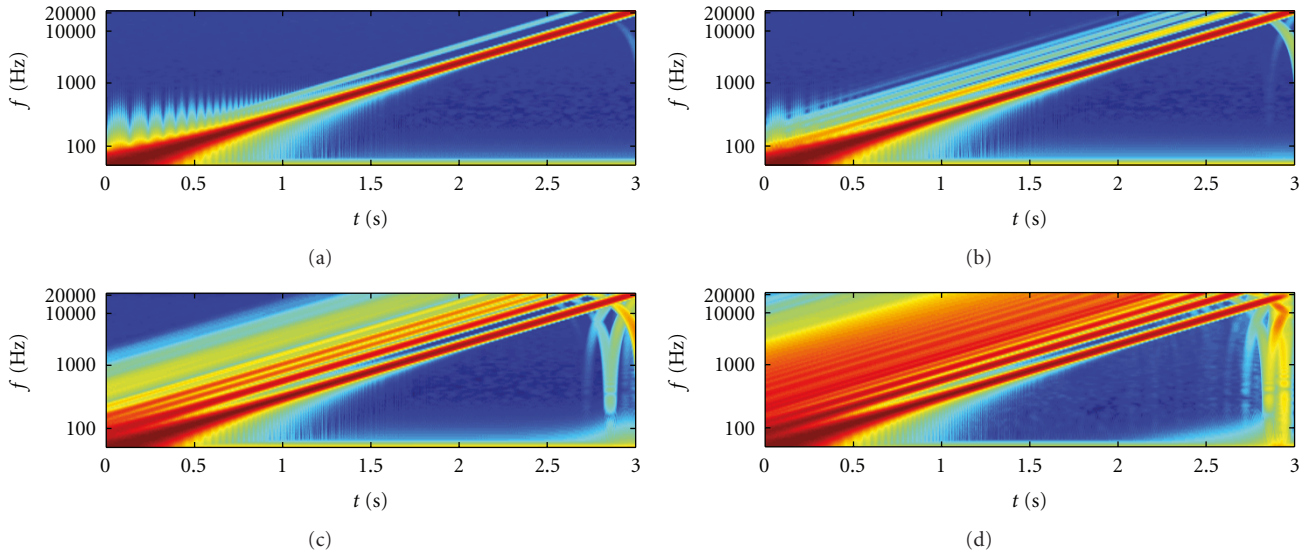
FIGURE 7: Simulation results for a logarithmic sweep signal. The plate voltage signals p_1, p_2, p_3, and p_4 are displayed. The 32-x oversampling was used to reduce aliasing.
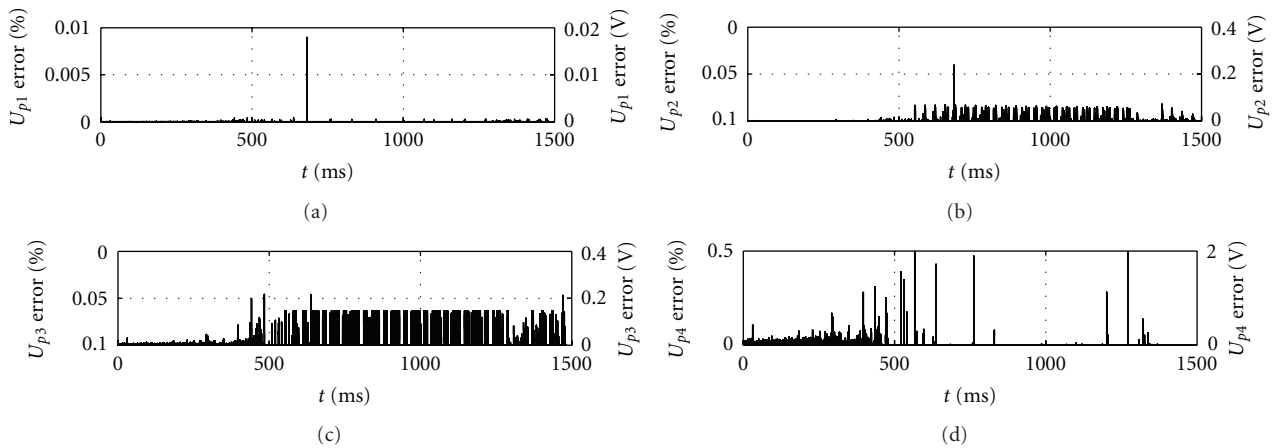

FIGURE 8: Comparison between simulation results using numerical solution and using approximations for a part of a real guitar riff. Only the error signals are displayed.
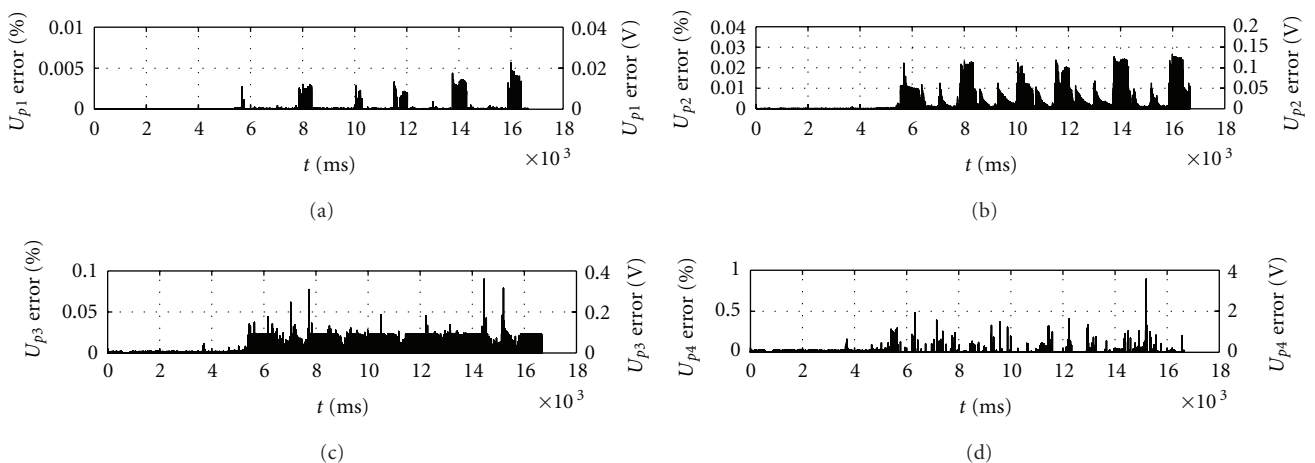

FIGURE 9: Comparison between simulation results using numerical solution and using approximations for the reduced power supply voltage.
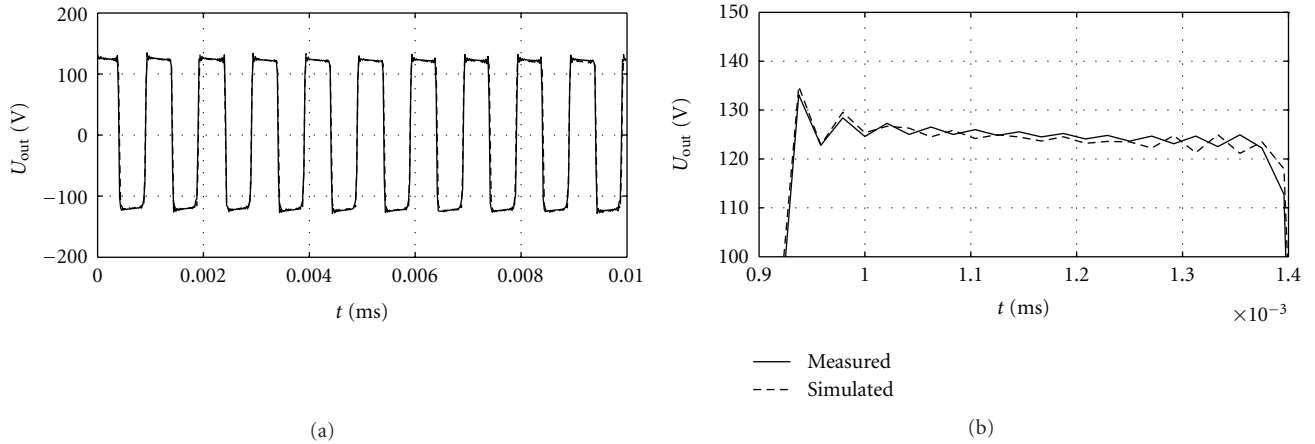
(a)

(b)

FIGURE 10: Comparison between measured and simulated preamp. The input voltage was sinewave signal with an amplitude of 150 mV and a frequency of 1 kHz.
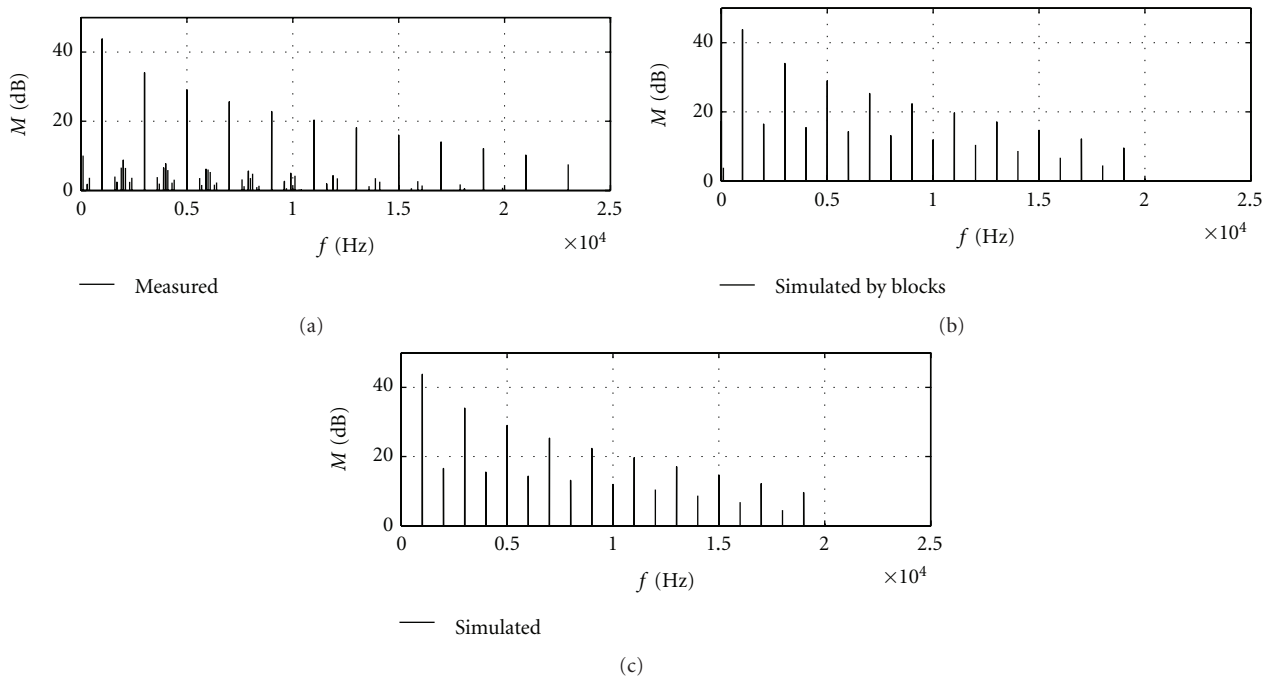

(a)

(b)

(c)

FIGURE 11: Spectrum comparison between measured and simulated preamps.

7. Conclusion

In this paper, real-time simulation of a guitar tube amplifier using approximations is proposed. The approximation of the solution of differential equations offers sufficient accuracy of the simulation while the computational cost is relatively low. The approximations are used together with the modified blockwise method that allows further reduction of the computational complexity. The blockwise method has been tested and it gives almost the same results as the simulation of the whole circuit. The results of the amplifier simulation were compared with the measurement of the real amplifier and the results show that the quite good accuracy of the simulation can be obtained. However, the compared signals differs in even harmonics. This was probably caused by the tube models that were used in simulation, because the numerical solution of the whole circuit and simulation using approximation were almost the same. Different amplification factor of the tube model can cause the bias shift resulting in different results.

The major advantage of the proposed algorithm is constant computational complexity and also the computational complexity is independent from the number of nonlinear functions in the simulated circuit or from the number of circuit nodes. It depends only on the number of accumulation elements. However, this is also disadvantageous, because the implementation of approximating functions is quite complicated. Therefore, practically, the number of

accumulation elements is bounded. Nevertheless, this disadvantage is compensated for by using the blockwise method. Compared to other methods, this method offers quite simple implementation, because it is derived directly from circuit equations. Thus, no transformations are necessary and the implementation should be faster. However, comparison with other methods has not been done yet and therefore this will be the next work.

Acknowledgment

References

[1] 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.

[2] 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 *Proceedings of the Digital Audio Effects (DAFx '07)*, pp. 197–204, Bordeaux, France, September 2007.

[3] D. T. Yeh, J. S. Abel, and J. O. Smith, "Simplified, physically-informed models of distortion and overdrive guitar effects pedal," in *Proceedings of the Digital Audio Effects (DAFx '07)*, pp. 189–196, Bordeaux, France, September 2007.

[4] A. Novak, L. Simon, and P. Lotton, "Analysis, synthesis, and classification of nonlinear systems using synchronized swept-sine method for audio effects," *EURASIP Journal on Advances in Signal Processing*, vol. 2010, Article ID 793816, 8 pages, 2010.

[5] J. Pakarinen, M. Tikander, and M. Karjalainen, "Wave digital modeling of the output chain of a vacuum-tube amplifier," in *Proceedings of the International Conference on Digital Audio Effects (DAFx '09)*, pp. 1–4, Como, Italy, September 2009.

[6] M. Karjalainen and J. Pakarinen, "Wave digital simulation of a vacuum-tube amplifier," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '06)*, pp. 153–156, Toulouse, France, May 2006.

[7] J. Pakarinen and M. Karjalainen, "Enhanced wave digital triode model for real-time tube amplifier emulation," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 18, no. 4, pp. 738–746, 2010.

[8] D. T. Yeh and J. O. Smith, "Simulating guitar distortion circuits using wave digital and nonlinear state-space formulations," in *Proceedings of the Digital Audio Effects (DAFx '08)*, pp. 19–26, Espoo, Finland, September 2008.

[9] K. Dempwolf, M. Holters, and U. ZöLzer, "Disceetization of parametric analog circuits for realtime simulations," in *Proceedings of the 13th International Conference on Digital Audio Effects (DAFx '10)*, Graz, Austria, September 2010.

[10] D. T. Yeh, J. S. Abel, and J. O. Smith, "Automated physical modeling of nonlinear audio circuits for real-time audio effectspart I: theoretical development," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 18, no. 4, pp. 728–737, 2010.

[11] I. Cohen and T. Helie, "Real-time simulation of a guitar power amplifier," in *Proceedings of the 13th International Conference on Digital Audio Effects (DAFx '10)*, Graz, Austria, September 2010.

[12] 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.

[13] J. Macak 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 (DAFx '10)*, Graz, Austria, September 2010.

[14] J. Macak, "Modified blockwise method for simulation of guitar tube amplifiers," in *Proceedings of 33nd International Conference Telecommunications and Signal Processing (TSP '10)*, pp. 1–4, Vienna, Austria, August 2010.

[15] N. Koren, "Improved vacuum tube models for SPICE simulations," 2003, http://www.normankoren.com/Audio/Tube-modspice_article.html.

[16] Spectrum Software, *Micro-Cap 10*, Spectrum Software, Sunnyvale, Calif, USA, 2010.

[17] 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*, pp. 367–385, Washington, DC, USA, September 1920.

[18] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C*, Cambridge University Press, Cambridge, UK, 2nd edition, 1992.

[19] U. ZöLzer, *DAFX—Digital Audio Effects*, John Wiley & Sons, New York, NY, USA, 1st edition, 2002.

[20] C. de Boor, *A Practical Guide to Splines*, Springer, New York, NY, USA, 1st edition, 2001.