

AUTOMATIC DECOMPOSITION OF NON-LINEAR EQUATION SYSTEMS IN AUDIO EFFECT CIRCUIT SIMULATION

Martin Holters, Udo Zölzer

Department of Circuits and Systems,
Helmut Schmidt University
University of the Federal Armed Forces
Hamburg, Germany
martin.holters|udo.zoelzer@hsu-hh.de

ABSTRACT

In the digital simulation of non-linear audio effect circuits, the arising non-linear equation system generally poses the main challenge for a computationally cheap implementation. As the computational complexity grows super-linearly with the number of equations, it is beneficial to decompose the equation system into several smaller systems, if possible. In this paper we therefore develop an approach to determine such a decomposition automatically. We limit ourselves to cases where an exact decomposition is possible, however, and do not consider approximate decompositions.

1. INTRODUCTION

Digital emulation of analog circuits for musical audio processing, like synthesizers, guitar effect pedals, or vintage amplifiers, is an ongoing research topic. Various methods exist to derive a mathematical model for an analog circuit in a systematic hence automatable way, most notably wave digital filters [1, 2, 3], port-Hamiltonian approaches [4, 5], and state-space based approaches [6, 7]. In the following, we will focus on the method of [7], as it has no limitations concerning the circuit topology and is general enough to also handle pathological circuits or element models (e.g. [8]). However, the underlying ideas should be equally applicable to the other approaches.

The downside of such automated approaches, and of [7] in particular, is that they often will lead to one large system of non-linear equations, collected from all the circuit’s non-linear elements. If possible, however, it is usually more efficient to solve many small equation systems instead of a single large one. Typically, the convergence of small systems will be better, allowing for a smaller number of iterations in an iterative solver. And more tangible, the complexity of solving the linear equation in e.g. the Newton method scales with the square of the number of involved equations, so that for a system of size N , the asymptotic complexity per iteration is $O(N^2)$, while for N systems of size 1, it is $O(N)$.

We therefore develop a method to decompose a non-linear equation system into smaller subsystems. We limit ourselves to the case where this is possible without resorting to approximations, although this unfortunately precludes the method from being applied to many circuits of practical relevance, especially those with global feedback paths. But while automatically deriving approximate decompositions, e.g. like those of [9, 10, 11], is beyond the scope of this paper, we are confident it is still useful by itself and furthermore, may form the basis for future methods to automatically find such approximate decompositions.

2. MODEL DERIVATION METHOD

We shall first provide a short introduction into the method used to obtain the circuit model and the non-linear equation in particular, focusing on the example of Figure 1 (based on “Der Birdie”¹), as also discussed in [12], while the reader is referred to [7] for details.

First, the equations of the individual circuit elements are rewritten in a uniform way in terms of branch voltages and currents and internal states. For example, a resistor with resistance R is described as

$$v_R + Ri_R = 0, \quad (1)$$

where v_R and i_R are the voltage across and the current through the resistor, respectively. The discrete-time model² of a capacitor with capacitance C is derived using bilinear transform as

$$\begin{pmatrix} C \\ 0 \end{pmatrix} v_C(n) + \begin{pmatrix} 0 \\ 1 \end{pmatrix} i_C(n) - \begin{pmatrix} \frac{1}{T} \\ \frac{1}{T} \end{pmatrix} x_C(n) = \begin{pmatrix} \frac{1}{2} \\ -\frac{1}{T} \end{pmatrix} x_C(n-1), \quad (2)$$

where T denotes the sampling interval and n the current time step, v_C and i_C are again the voltage across and the current through the capacitor, respectively, and the state x_C corresponds to the capacitor’s charge. Voltage sources are easily expressed using a non-zero right-hand side, e.g. as

$$v_{V_{CC}} + 0 \cdot i_{V_{CC}} = 9 \text{ V} \quad (3)$$

for the supply voltage source V_{CC} , with a constant voltage $v_{V_{CC}} = 9 \text{ V}$ across it, driving an arbitrary current $i_{V_{CC}}$.

The non-linear equation of non-linear elements is expressed in terms of an auxiliary vector \mathbf{q} , which is related to voltages and currents (and potentially also states) through a linear equation. Thus, a diode is expressed using the linear equation

$$\begin{pmatrix} 1 \\ 0 \end{pmatrix} v_D(n) + \begin{pmatrix} 0 \\ 1 \end{pmatrix} i_D(n) + \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix} \mathbf{q}_D(n) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad (4)$$

fixing $q_{D,1} = v_D$ as the voltage across and $q_{D,2} = i_D$ as the current through the diode, and the non-linear Shockley equation, rewritten to implicit form as

$$f_D(\mathbf{q}_D) = I_s \cdot \left(e^{\frac{q_{D,1}}{\eta v_T}} - 1 \right) - q_{D,2} = 0, \quad (5)$$

¹<http://diy.musikding.de/wp-content/uploads/2013/06/birdieschalt.pdf>

²In [7], the equations are first derived in the continuous-time domain and then transformed to discrete time using the bilinear form. Equivalently, the bilinear transform may be applied per element, which we do here for brevity’s sake.

where I_s , η , and v_T denote, respectively, reverse saturation current, emission coefficient (ideality factor), and thermal voltage. Similarly, a transistor is expressed using the linear equation

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{pmatrix} \mathbf{v}_T(n) + \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 1 & 0 \end{pmatrix} \mathbf{i}_T(n) + \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} \mathbf{q}_T(n) = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad (6)$$

where \mathbf{v}_T and \mathbf{i}_T hold the voltages across and currents through the base-emitter and base-collector branch in that order, respectively, and the non-linear equation is obtained by rewriting the Ebers-Moll equation to implicit form as

$$f_T(\mathbf{q}_T) = \begin{pmatrix} I_{sE} \cdot (e^{\frac{q_{T,1}}{\eta_E v_T}} - 1) - \frac{\beta_r}{1+\beta_r} I_{sC} \cdot (e^{\frac{q_{T,2}}{\eta_C v_T}} - 1) - q_{T,3} \\ -\frac{\beta_f}{1+\beta_f} I_{sE} \cdot (e^{\frac{q_{T,1}}{\eta_E v_T}} - 1) + I_{sC} \cdot (e^{\frac{q_{T,2}}{\eta_C v_T}} - 1) - q_{T,4} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad (7)$$

where β_f and β_r denote forward and reverse current gain, respectively, and the reverse saturation currents I_{sE} and I_{sC} and the emission coefficients η_E and η_C can differ between base-emitter and base-collector junction.

Once the individual elements are modeled in a suitable form, the circuit topology is taken into consideration by formulating the Kirchhoff voltage law

$$\mathbf{T}_v \mathbf{v} = \mathbf{0} \quad (8)$$

and the Kirchhoff current law

$$\mathbf{T}_i \mathbf{i} = \mathbf{0}, \quad (9)$$

where \mathbf{v} and \mathbf{i} collect all the circuit's branch voltages and currents, using matrices \mathbf{T}_v of independent loop and \mathbf{T}_i of independent node (or cut-set) equations, which can be obtained by well-known methods (see e.g. [13]). For the circuit of Figure 1 one e.g. finds

$$\mathbf{T}_v = \begin{pmatrix} V_{CC} & C_5 & D & V_{in} & R_1 & C_1 & R_2 & R_3 & T_{BE} & T_{BC} & R_4 & R_5 & C_3 & P_{1,1} & P_{1,2} \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 1 & 1 \end{pmatrix} \quad (10)$$

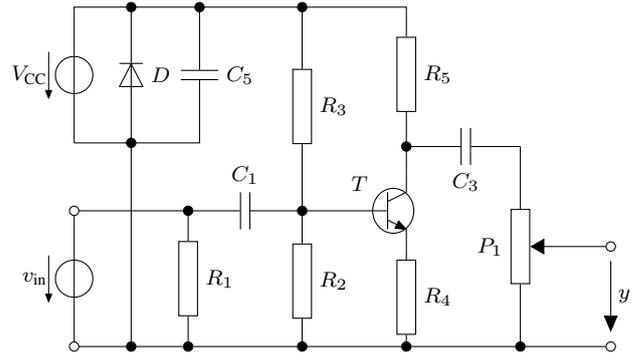


Figure 1: Example treble booster circuit.

Table 1: Component/parameter values for the circuit of Figure 1.

comp.	value	comp.	param.	value
V_{CC}	9 V	D	I_s	350 pA
R_1	1 M Ω	D	η	1.6
R_2	43 k Ω	T	I_{sE}	64.53 fA
R_3	430 k Ω	T	I_{sC}	154.1 fA
R_4	390 Ω	T	η_E	1.06
R_5	10 k Ω	T	η_C	1.10
P_1	100 k Ω	T	β_f	500
C_1	2.2 nF	T	β_r	12
C_3	2.2 nF			
C_5	100 μ F			

and

$$\mathbf{T}_i = \begin{pmatrix} V_{CC} & C_5 & D & V_{in} & R_1 & C_1 & R_2 & R_3 & T_{BE} & T_{BC} & R_4 & R_5 & C_3 & P_{1,1} & P_{1,2} \\ 1 & -1 & -1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & -1 & 0 & 0 & -1 & -1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{pmatrix}. \quad (11)$$

Combining the element equations with the topology equations yields the equation system

$$\begin{pmatrix} \mathbf{M}_v & \mathbf{M}_i & \mathbf{M}_x & \mathbf{M}_q \\ \mathbf{T}_v & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{T}_i & \mathbf{0} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{v}(n) \\ \mathbf{i}(n) \\ \mathbf{x}(n) \\ \mathbf{q}(n) \end{pmatrix} = \begin{pmatrix} \mathbf{M}_x \\ \mathbf{0} \\ \mathbf{0} \end{pmatrix} \mathbf{x}(n-1) + \begin{pmatrix} \mathbf{M}_u \\ \mathbf{0} \\ \mathbf{0} \end{pmatrix} \mathbf{u}(n) + \begin{pmatrix} \mathbf{u}_0 \\ \mathbf{0} \\ \mathbf{0} \end{pmatrix}. \quad (12)$$

The matrices \mathbf{M}_v , \mathbf{M}_i , \mathbf{M}_x , \mathbf{M}_q , and \mathbf{M}_x , are constructed as block diagonal matrices (with potentially rectangular blocks) from the respective factors in the element equations, and \mathbf{x} and \mathbf{q} are obtained by stacking the respective entries of the individual elements. For the example circuit, the top left parts of the matrices, corresponding

to V_{CC} , C_5 , and D , e.g. are

$$\mathbf{M}_v = \begin{pmatrix} 1 & 0 & 0 & \dots \\ 0 & C_5 & 0 & \dots \\ 0 & 0 & 0 & \dots \\ 0 & 0 & 1 & \dots \\ 0 & 0 & 0 & \dots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix} \quad \mathbf{M}_i = \begin{pmatrix} 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & \dots \\ 0 & 1 & 0 & \dots \\ 0 & 0 & 0 & \dots \\ 0 & 0 & 1 & \dots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix} \quad (13)$$

$$\mathbf{M}_x = \begin{pmatrix} 0 & \dots \\ -\frac{1}{2} & \dots \\ -\frac{1}{T} & \dots \\ 0 & \dots \\ 0 & \dots \\ \vdots & \ddots \end{pmatrix} \quad \mathbf{M}_z = \begin{pmatrix} 0 & \dots \\ \frac{1}{2} & \dots \\ -\frac{1}{T} & \dots \\ 0 & \dots \\ 0 & \dots \\ \vdots & \ddots \end{pmatrix} \quad (14)$$

$$\mathbf{M}_q = \begin{pmatrix} 0 & 0 & \dots \\ 0 & 0 & \dots \\ 0 & 0 & \dots \\ -1 & 0 & \dots \\ 0 & -1 & \dots \\ \vdots & \vdots & \ddots \end{pmatrix}. \quad (15)$$

The vector \mathbf{u}_0 holds constant source values, in this case a single non-zero as first entry, namely $V_{CC} = 9\text{V}$, while $\mathbf{u}(n)$ holds all time-varying source values, i.e. the circuit inputs, in this case only the input voltage $v_{in}(n)$, which is mapped to the appropriate row by the matrix \mathbf{M}_q , a one-column matrix in the example with a single 1 in the sixth row.

The matrix on the left-hand side of (12) has dimension 36×39 , hence the equation system is under-determined. However, we can solve for a solution set

$$\begin{pmatrix} \mathbf{v}(n) \\ \mathbf{i}(n) \\ \mathbf{q}(n) \end{pmatrix} = \begin{pmatrix} \mathbf{D}_v \\ \mathbf{D}_i \\ \mathbf{A} \\ \mathbf{D}_q \end{pmatrix} \mathbf{x}(n-1) + \begin{pmatrix} \mathbf{E}_v \\ \mathbf{E}_i \\ \mathbf{B} \\ \mathbf{E}_q \end{pmatrix} \mathbf{u}(n) + \begin{pmatrix} \mathbf{v}_0 \\ \mathbf{i}_0 \\ \mathbf{x}_0 \\ \mathbf{q}_0 \end{pmatrix} + \begin{pmatrix} \mathbf{F}_v \\ \mathbf{F}_i \\ \mathbf{C} \\ \mathbf{F}_q \end{pmatrix} \mathbf{z}(n), \quad (16)$$

where the first three terms form a particular solution, while the last term with a three-dimensional arbitrary vector $\mathbf{z}(n)$ spans the solution set. Note that neither the particular solution nor the basis of the nullspace to span the solution set are unique. In the ACME framework³ that implements this method, an algorithm based on [14] is used that exploits the sparsity of the matrices, and the particular solution is further modified such that \mathbf{D}_q and \mathbf{E}_q are orthogonal to \mathbf{F}_q as motivated in [12].

In the following, the last row of (16) is of particular interest, and we shall drop the time index n , i.e. we write

$$\mathbf{q} = \mathbf{D}_q \mathbf{x} + \mathbf{E}_q \mathbf{u} + \mathbf{q}_0 + \mathbf{F}_q \mathbf{z}. \quad (17)$$

In order to find \mathbf{z} , the non-linear element equations need to be

considered, which are gathered to obtain

$$\begin{aligned} \mathbf{f}(\mathbf{q}) &= \mathbf{f} \left(\begin{pmatrix} \mathbf{q}_D \\ \mathbf{q}_T \end{pmatrix} \right) = \begin{pmatrix} \mathbf{f}_D(\mathbf{q}_D) \\ \mathbf{f}_T(\mathbf{q}_T) \end{pmatrix} \\ &= \begin{pmatrix} I_s \cdot (e^{\frac{q_1}{\eta v_T}} - 1) - q_2 \\ I_{sE} \cdot (e^{\frac{q_3}{\eta_E v_T}} - 1) - \frac{\beta_r}{1+\beta_r} I_{sC} \cdot (e^{\frac{q_4}{\eta_C v_T}} - 1) - q_5 \\ -\frac{\beta_r}{1+\beta_r} I_{sE} \cdot (e^{\frac{q_3}{\eta_E v_T}} - 1) + I_{sC} \cdot (e^{\frac{q_4}{\eta_C v_T}} - 1) - q_6 \end{pmatrix} = \mathbf{0}. \end{aligned} \quad (18)$$

Note that the number $N_n = 3$ of subequations is smaller than the number $N_q = 6$ of entries in the vector \mathbf{q} . Thus, this implicit non-linear equation confines \mathbf{q} to a solution manifold. On the other hand, \mathbf{q} is restricted to the affine subspace spanned by (17) for arbitrary \mathbf{z} . As \mathbf{z} has exactly $N_n = 3$ entries, there is in general a finite number of permissible \mathbf{z} , and for physically meaningful circuit schematics, there will be a unique solution. Once \mathbf{z} is found, it is used to calculate the circuit's output (by extracting the desired entry of $\mathbf{v}(n)$) and update its states according to (16).

3. DECOMPOSITION METHOD

In general, the non-linear equation system is formed by collecting subequations of the N individual non-linear elements contained in the circuit as

$$\mathbf{f}(\mathbf{q}) = \begin{pmatrix} \mathbf{f}_1(\mathbf{q}_1) \\ \mathbf{f}_2(\mathbf{q}_2) \\ \vdots \\ \mathbf{f}_N(\mathbf{q}_N) \end{pmatrix} = \mathbf{0} \quad (19)$$

where \mathbf{q} is likewise formed from subvectors as

$$\mathbf{q}^T = (\mathbf{q}_1^T \quad \mathbf{q}_2^T \quad \dots \quad \mathbf{q}_N^T). \quad (20)$$

Unfortunately, this does not mean that the subequations can be solved individually, as we need to solve for \mathbf{z} , not \mathbf{q} or the individual \mathbf{q}_n . In fact, it will only be possible in very benign cases to solve the equations element-by-element. It is more likely that groups of elements can be identified into which the non-linear equation can be decomposed.

3.1. Decomposition assuming known grouping

Assume the non-linear equation system is split into M equation groups $\tilde{\mathbf{f}}_m(\tilde{\mathbf{q}}_m)$, $m = 1, \dots, M$ with $M \leq N$. That is, each $\tilde{\mathbf{q}}_m$ and $\tilde{\mathbf{f}}_m(\tilde{\mathbf{q}}_m)$ is the concatenation of one or more \mathbf{q}_n and $\mathbf{f}_n(\mathbf{q}_n)$, respectively. As the ordering of the elements in $\mathbf{f}(\mathbf{q})$ and \mathbf{q} is arbitrary, we may assume without loss of generality that

$$\mathbf{f}^T(\mathbf{q}) = (\tilde{\mathbf{f}}_1^T(\tilde{\mathbf{q}}_1) \quad \dots \quad \tilde{\mathbf{f}}_M^T(\tilde{\mathbf{q}}_M)) \quad (21)$$

and

$$\mathbf{q}^T = (\tilde{\mathbf{q}}_1^T \quad \dots \quad \tilde{\mathbf{q}}_M^T). \quad (22)$$

Let $\mathbf{D}_{q,m}$, $\mathbf{E}_{q,m}$, and $\mathbf{F}_{q,m}$ denote the corresponding rows of the respective matrices and likewise $\mathbf{q}_{0,m}$ the corresponding entries of \mathbf{q}_0 so that

$$\tilde{\mathbf{q}}_m = \mathbf{D}_{q,m} \mathbf{x} + \mathbf{E}_{q,m} \mathbf{u} + \mathbf{q}_{0,m} + \mathbf{F}_{q,m} \mathbf{z}. \quad (23)$$

Further let \mathbf{z} also be decomposed as

$$\mathbf{z}^T = (\mathbf{z}_1^T \quad \mathbf{z}_2^T \quad \dots \quad \mathbf{z}_M^T) \quad (24)$$

³<https://github.com/HSU-ANT/ACME.jl>

such that \mathbf{z}_m has as many entries as $\tilde{\mathbf{f}}_m(\tilde{\mathbf{q}}_m)$ and let $\mathbf{F}_{q,m,n}$ denote the corresponding columns of $\mathbf{F}_{q,m}$, that is

$$\mathbf{F}_q = \begin{pmatrix} \mathbf{F}_{q,1,1} & \cdots & \mathbf{F}_{q,1,M} \\ \vdots & \ddots & \vdots \\ \mathbf{F}_{q,M,1} & \cdots & \mathbf{F}_{q,M,M} \end{pmatrix} \quad (25)$$

such that

$$\tilde{\mathbf{q}}_m = \mathbf{D}_{q,m}\mathbf{x} + \mathbf{E}_{q,m}\mathbf{u} + \mathbf{q}_{0,m} + \mathbf{F}_{q,m,1}\mathbf{z}_1 + \cdots + \mathbf{F}_{q,m,M}\mathbf{z}_M. \quad (26)$$

Now, let the element groups be chosen such that $\mathbf{F}_{q,m,n} = \mathbf{0}$ for $n > m$. Then

$$\tilde{\mathbf{q}}_1 = \mathbf{D}_{q,1}\mathbf{x} + \mathbf{E}_{q,1}\mathbf{u} + \mathbf{q}_{0,1} + \mathbf{F}_{q,1,1}\mathbf{z}_1 \quad (27)$$

so that \mathbf{z}_1 can be obtained by solving $\tilde{\mathbf{f}}_1(\tilde{\mathbf{q}}_1) = \mathbf{0}$. With \mathbf{z}_1 known, \mathbf{z}_2 can then be obtained by solving $\tilde{\mathbf{f}}_2(\tilde{\mathbf{q}}_2) = \mathbf{0}$ and so forth up to M . Thus, if \mathbf{F}_q is such that a partitioning with $\mathbf{F}_{q,m,n} = \mathbf{0}$ for $n > m$ (and $M > 1$) exists, we can decompose the non-linear equation into individually solvable subequations, where in general, the m -th subequation depends on the solution of the $m-1$ previous subequations.

Of course, we may rarely be lucky and find \mathbf{F}_q to be suitable for this decomposition. However, [7] leaves some freedom in the exact choice of the coefficient matrices. In particular, only the space spanned by $\mathbf{F}_q\mathbf{z}$ is of importance, which does not change if we substitute $\mathbf{F}_q \leftarrow \mathbf{F}_q\mathbf{R}$ for a regular matrix \mathbf{R} . (This will, of course, change the resulting \mathbf{z} , so \mathbf{F}_v , \mathbf{F}_i , and \mathbf{C} have to be updated in the same way.) Now, finding an \mathbf{R} such that the upper right subblocks of \mathbf{F}_q become zero is similar to bringing \mathbf{F}_q^T into upper triangular form, but with respect to the rectangular subblocks $\mathbf{F}_{q,m,n}$. Thus if the chosen decomposition into subequation groups allows a suitable choice of \mathbf{F}_q , it can be found using standard tools of linear algebra, e.g. Gaussian elimination.

3.2. Identification of a suitable grouping

The remaining question is how to determine a suitable grouping. If \mathbf{f}_q and \mathbf{q} were ordered in correspondence with the yet-to-be-found grouping, i.e. fulfilling (21) and (22), we could just determine \mathbf{R} to eliminate the maximum number of elements in the upper right part of \mathbf{F}_q and examine the zero-structure thus obtained. Unfortunately, the number of possible permutations of the entries in \mathbf{f}_q and \mathbf{q} grows too fast with N to make trying all of them feasible. E.g. for $N = 10$ non-linear elements, we would need to examine $N! = 3\,628\,800$ permutations.

We can do a little better than that by greedily trying to separate a single subgroup by trying all $2^N - 1$ non-empty subsets of $\{1, \dots, N\}$ in order of increasing cardinality. Once the smallest permissible subgroup has been identified, i.e. one for which \mathbf{F}_q can be transformed in a suitable way when that subgroup is placed first in \mathbf{f}_q and \mathbf{q} , the process is repeated for the remaining elements. In the worst case, if the circuit does not allow decomposition, all $2^N - 1$ non-empty subsets have to be tried. Otherwise, the number of trials in the first iteration is lower, but additional trials are needed for the remaining non-linear elements. Nevertheless, it can be seen that the complete procedure never has to try more than $2^N - 1$ subgroupings. This is still an exponential growth with N , but in the realm of audio effect circuits where a number N of non-linear elements in the low two-digit range is already considered quite complex, the needed computational time during the offline

pre-computation step may be well acceptable. E.g. for the $N = 10$ case, at most 1023 trials would be needed.

3.3. Dimensionality reduction of the input vector

After the decomposition, we can obtain \mathbf{z}_m from $\tilde{\mathbf{f}}_m(\tilde{\mathbf{q}}_m) = \mathbf{0}$, which depends on \mathbf{x} , \mathbf{u} , and $\mathbf{z}_1, \dots, \mathbf{z}_{m-1}$, a potentially high number of input values. This would pose a major problem if one would like to tabulate precomputed values in a look-up table. However, the method proposed in [12] can be easily extended to not only treat \mathbf{x} and \mathbf{u} as inputs, but also $\mathbf{z}_1, \dots, \mathbf{z}_{m-1}$, to find an index vector \mathbf{p}_m of minimal dimension that can be used as input.

The idea is to apply a rank factorization to the matrix

$$\begin{pmatrix} \mathbf{D}_{q,m} & \mathbf{E}_{q,m} & \mathbf{F}_{q,m,1} & \cdots & \mathbf{F}_{q,m,m-1} \end{pmatrix} = \mathbf{Q}_m \cdot \begin{pmatrix} \hat{\mathbf{D}}_{q,m} & \hat{\mathbf{E}}_{q,m} & \hat{\mathbf{F}}_{q,m,1} & \cdots & \hat{\mathbf{F}}_{q,m,m-1} \end{pmatrix} \quad (28)$$

such that $\begin{pmatrix} \hat{\mathbf{D}}_{q,m} & \hat{\mathbf{E}}_{q,m} & \hat{\mathbf{F}}_{q,m,1} & \cdots & \hat{\mathbf{F}}_{q,m,m-1} \end{pmatrix}$ has minimal number of rows. Then

$$\mathbf{p}_m = \hat{\mathbf{D}}_{q,m}\mathbf{x} + \hat{\mathbf{E}}_{q,m}\mathbf{u} + \hat{\mathbf{F}}_{q,m,1}\mathbf{z}_1 + \cdots + \hat{\mathbf{F}}_{q,m,m-1}\mathbf{z}_{m-1} \quad (29)$$

is the index vector of minimal dimension to be used in

$$\tilde{\mathbf{q}}_m = \mathbf{q}_{0,m} + \mathbf{Q}_m\mathbf{p}_m + \mathbf{F}_{q,m,m}\mathbf{z}_m. \quad (30)$$

4. EXAMPLES

4.1. Treble booster

As a first, relatively trivial example, we consider the treble booster circuit of Figure 1 with the component values given in Table 1. The circuit contains two non-linear elements, a diode and a transistor. But note that the formers sole purpose is to protect the circuit against connecting the power supply with wrong polarity. Usually, one would omit the diode from the simulation as it has no influence on the output signal. Here, we include it to verify that we can then eliminate it algorithmically.

We choose to put the diode first, so that \mathbf{q}_1 has two elements and $\mathbf{f}_1(\mathbf{q}_1)$ has one (see (5)), and \mathbf{q}_2 has four elements and $\mathbf{f}_2(\mathbf{q}_2)$ has two (see (7)). Using the ACME implementation of [7], we find

$$\mathbf{F}_q = \left(\begin{array}{c|cc} 0 & 0 & 0 \\ 1 & 0 & 0 \\ \hline 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -2.917 \times 10^{-4} & 9.705 \times 10^{-5} \\ 0 & 9.705 \times 10^{-5} & -1.054 \times 10^{-4} \end{array} \right) \left. \begin{array}{l} \\ \\ \\ \\ \end{array} \right\} \begin{array}{l} D_1 \\ \\ T_1 \end{array} \quad (31)$$

where the lines demark the decomposition according to (25).

Apparently, $\mathbf{F}_{q,1,2} = \mathbf{0}$ without any further modifications, and we can determine \mathbf{z}_1 from the diode equation alone. In this particular case, we do not even need \mathbf{z}_1 to determine \mathbf{z}_2 , as also $\mathbf{F}_{q,2,1} = \mathbf{0}$, so we could just as well have put the diode second.

Using the method of [12] to determine index vectors \mathbf{p}_m of minimal dimension, we find \mathbf{p}_2 has two entries, while \mathbf{p}_1 has no entries at all. Thus, $\mathbf{f}_1(\mathbf{q}_1)$ and hence also \mathbf{z}_1 do not depend on values changing during simulation and can therefore be pre-computed offline.

To evaluate the performance impact, a guitar signal of 33.6 s duration sampled at 44.1 kHz is processed with the help of ACME

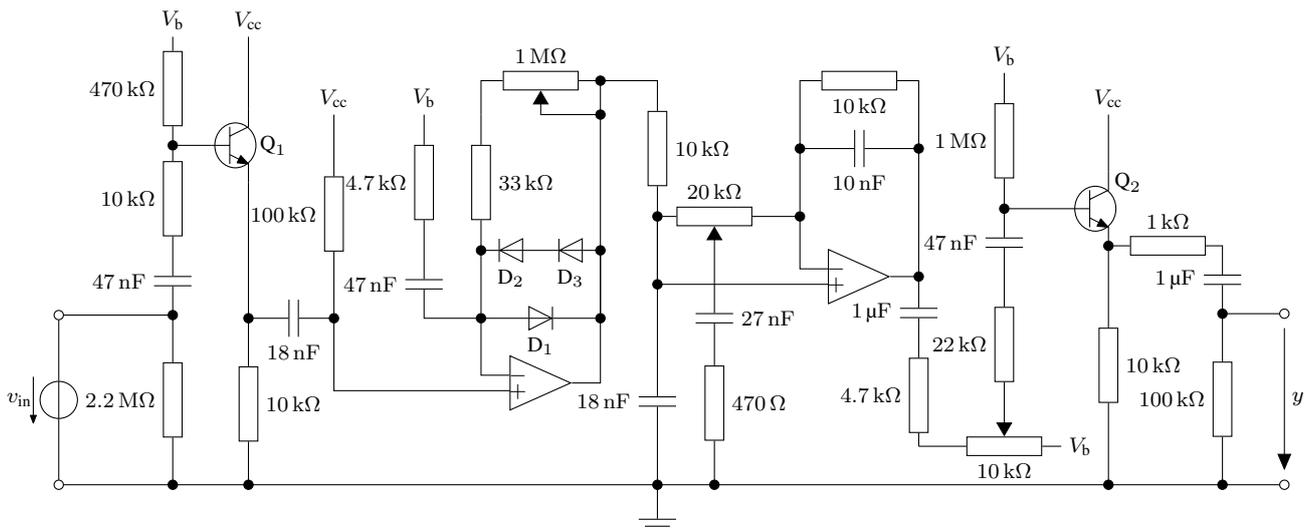


Figure 2: Example overdrive circuit.

v0.4.1 and the required processing time is measured using Julia v0.6.0 on an Intel Xeon E5-1620v2 CPU at 3.7 GHz. The non-linear equation is solved using Newton’s method, where the initial solution is determined by extrapolating from the previous time step’s solution using linearization (see [12] for details). No further pre-computing/caching of solutions is employed. As the circuit introduces relatively little distortion, only 1.789 iterations are required on average to refine the initial solution. While the original model needs 2.45 s to run, the decomposed one reduces the time to 1.99 s, an improvement by 19 %.

4.2. Overdrive

As a second example, we consider the more complex overdrive circuit of Figure 2 (based on “Der Super Over”⁴). Again, there is a diode anti-parallel to the supply voltage source as shown in Figure 3a which we include in the model. However, we simplify the bias voltage V_b generation; while the original circuit contains a voltage divider and a stabilizing capacitor (see Figure 3b), we enforce a constant bias voltage by directly connecting an ideal voltage source (see Figure 3c). We assume the operational amplifiers ideal, leading to a model with six non-linear elements: the protective diode anti-parallel to the supply voltage, three diodes in the feedback around the first operational amplifier, and two transistors. The non-linear equation system $\mathbf{f}(\mathbf{q}) = \mathbf{0}$ therefore comprises eight equations (one per diode, two per transistor), while vector \mathbf{q} has 16 entries (two per diode, four per transistor).

We again choose to put the protective diode first, then proceed

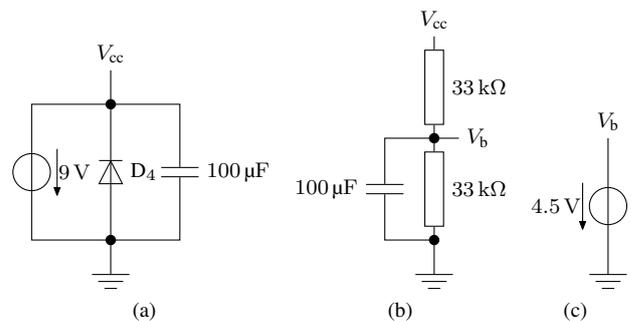


Figure 3: Power supply circuitry omitted from Figure 2 for (a) main supply voltage, (b) bias voltage, (c) simplified bias voltage.

left to right, and the ACME implementation of [7] yields

$$\mathbf{F}_q = \begin{pmatrix}
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
 0 & * & * & 0 & 0 & 0 & 0 & 0 \\
 0 & * & * & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & -1 & -1 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
 0 & * & * & * & 0 & 1 & 0 & 0 \\
 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
 0 & 0 & 0 & 0 & 0 & 0 & * & * \\
 0 & * & * & * & 0 & 0 & * & *
 \end{pmatrix} \begin{matrix}
 \left. \begin{matrix} D_4 \\ Q_1 \end{matrix} \right\} \\
 \left. \begin{matrix} D_2 \\ D_1 \end{matrix} \right\} \\
 \left. \begin{matrix} D_3 \\ Q_2 \end{matrix} \right\}
 \end{matrix} \quad (32)$$

where * denotes non-zero entries whose exact values depend on the potentiometer settings. It can be observed that the first diode

⁴<http://diy.musikding.de/wp-content/uploads/2013/06/superoverschalt.pdf>

can again be extracted without problems. Continuing by just considering the remaining matrix, the first transistor can likewise be extracted without the need to modify F_q . Now looking at the remaining submatrix

$$\left(\begin{array}{ccc|ccc} -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ \hline 1 & 0 & 0 & 0 & 0 & 0 \\ * & 0 & 1 & 0 & 0 & 0 \\ \hline 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & * & * & * \\ * & 0 & 0 & * & * & * \end{array} \right) \left. \begin{array}{l} \vphantom{\begin{array}{c} \\ \\ \\ \\ \\ \\ \\ \\ \\ \end{array}} \right\} D_2 \\ \left. \vphantom{\begin{array}{c} \\ \\ \\ \\ \\ \\ \\ \\ \\ \end{array}} \right\} D_1 \\ \left. \vphantom{\begin{array}{c} \\ \\ \\ \\ \\ \\ \\ \\ \\ \end{array}} \right\} D_3 \\ \left. \vphantom{\begin{array}{c} \\ \\ \\ \\ \\ \\ \\ \\ \\ \end{array}} \right\} Q_2$$

for the three diodes and the second transistor, none of the three diodes can be extracted by itself. Each of the three row pairs belonging to the three diodes obviously forms a matrix of rank 2, so we cannot possibly find a regular matrix with which to multiply from the right to zero out all but one of the columns. Likewise, the four lowest rows, belonging to the transistor, obviously form a rank-3 matrix, from which we cannot cancel all but two columns. As no single element can be extracted, the next thing to try is to extract pairs of elements. Again, none of the six possible pairs turn out to be extractable. But continuing with groups of three, the three diodes can be extracted as one group, without requiring modifications of F_q .

We thus arrive at the decomposition

$$F_q = \left(\begin{array}{ccc|ccc|ccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & * & * & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & * & * & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & * & * & * & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & * & * & * \\ 0 & * & * & * & 0 & 0 & * & * & * \end{array} \right) , \quad (33) \left. \begin{array}{l} \vphantom{\begin{array}{c} \\ \\ \\ \\ \\ \\ \\ \\ \\ \end{array}} \right\} D_4 \\ \left. \vphantom{\begin{array}{c} \\ \\ \\ \\ \\ \\ \\ \\ \\ \end{array}} \right\} Q_1 \\ \left. \vphantom{\begin{array}{c} \\ \\ \\ \\ \\ \\ \\ \\ \\ \end{array}} \right\} D_2 \\ \left. \vphantom{\begin{array}{c} \\ \\ \\ \\ \\ \\ \\ \\ \\ \end{array}} \right\} D_1 \\ \left. \vphantom{\begin{array}{c} \\ \\ \\ \\ \\ \\ \\ \\ \\ \end{array}} \right\} D_3 \\ \left. \vphantom{\begin{array}{c} \\ \\ \\ \\ \\ \\ \\ \\ \\ \end{array}} \right\} Q_2$$

comprising four subsystems of one, two, three, and two equations, respectively. Unlike the first example, the sub-diagonal blocks do contain non-zero entries, so the solution of earlier equations is required for the later ones (with the exception of the protective diode). Looking at the index vectors obtained with the method of [12], p_1 again has no entries, so the solution to $\tilde{f}_1(\tilde{q}_1) = \mathbf{0}$ can be pre-computed off-line. Furthermore, p_2 and p_4 both have two entries and p_3 just one, greatly simplifying the construction of lookup tables compared to a p with five entries for the original (not decomposed) system.

Performing the same performance evaluation as for the treble booster example above, the processing time of the original model is determined as 6.82 s with 2.965 Newton iterations needed on average per sample. Decomposing the model reduces the number

of iterations needed to 1.460, 2.924 and 1.696 for the three sub-equations, respectively. The needed processing time however is reduced insignificantly to 6.79 s. Nevertheless, the advantage of making lookup tables feasible remains.

Unfortunately, the obtained results depend on the simplification of fixing the bias voltage V_b . With the original circuitry, all nodes connected to V_b influence each other, effectively creating a global feedback path, and the only decomposition possible is the extraction of the protective diode of the main power supply. In terms of the F_q matrix, this manifests itself in additional entries in the fourth and last column that cannot be canceled and preclude further decomposition.

5. IMPLEMENTATION ASPECTS

The proposed method has been implemented as part of the ACME framework. One of the main challenges encountered is the condition $F_{q,m,n} = \mathbf{0}$ which, when using floating point arithmetic, will usually only be fulfilled approximately. The obvious approach then is to treat entries with very small absolute value as zero. However, determining an appropriate threshold proves to be anything but trivial, as the scale of the entries in both q and z can differ by orders of magnitude from each other. So instead, we employ exact arithmetic, using rationals of arbitrary precision integers. This is possible as the whole model derivation process only needs a bounded number of additions, subtractions, multiplications, and divisions, so that the numerators and denominators may grow very large, but are still bounded. Of course, for the simulation itself, the values are converted to floating point for efficiency.

Unfortunately, the benefits reaped from the decomposition in terms of run-time turn out to be smaller than expected; sometimes, the decomposed model may even run slightly slower than the original one. The reason seems to be constant overhead that may dominate over the asymptotic behavior for small problem sizes. In an earlier implementation, where LAPACK routines were used for linear equation solving, this effect was even more pronounced. The constant calling overhead for the LAPACK routines is significant: Solving a system of size eight takes less than five times as long as solving a system of size one. We hope for a future, optimized implementation to further reduce these overheads, however.

6. CONCLUSION

The proposed method is able to decompose a system of non-linear equations into a number of smaller subsystems if that is possible in an exact way. This may lead to more efficient simulations in terms of computational load (when using iterative solvers) or memory requirements (when using lookup tables). However, the time needed to find this decomposition during model creation grows exponentially with the number of non-linear circuit elements. Fortunately, typically modeled audio effect circuits do not contain enough non-linear elements to make the approach infeasible.

The present paper does not, however, tackle the more challenging problem of automatically finding an approximate decomposition, which may either give a solution of sufficient accuracy for the complete system directly, or could at least be used to find a good initial solution for an iterative solver then applied to the complete system. We hope this paper to be a valuable step in that direction, however. E.g. for the overdrive circuit, numerical analysis could reveal that the bias voltage is almost constant, and that in fact fixing it at a constant does not change the output in a significant way. Based

on that, the proposed method could be applied as exemplified in Sec. 4.2.

7. REFERENCES

- [1] G. De Sanctis and A. Sarti, “Virtual analog modeling in the wave-digital domain,” *IEEE Trans. on Audio, Speech and Language Process.*, vol. 18, no. 4, pp. 715–727, 2010.
- [2] A. Bernardini, K. J. Werner, A. Sarti, and J. O. Smith, “Modeling a class of multi-port nonlinearities in wave digital structures,” in *23rd European Signal Process. Conf. (EUSIPCO)*, Nice, France, 2015, pp. 664–668.
- [3] K. Werner, V. Nangia, J. Smith, and J. Abel, “Resolving wave digital filters with multiple/multiport nonlinearities,” in *Proc. 18th Int. Conf. Digital Audio Effects (DAFx-15)*, Trondheim, Norway, 2015.
- [4] A. Falaize and T. Hélie, “Passive guaranteed simulation of analog audio circuits: A port-hamiltonian approach,” *Applied Sciences*, vol. 6, no. 10, 2016.
- [5] A. Falaize-Skrzek and T. Hélie, “Simulation of an analog circuit of a wah pedal: a port-hamiltonian approach,” in *135th Audio Engineering Society Convention*, New York, 2013.
- [6] 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. on Audio, Speech and Language Process.*, vol. 18, no. 4, pp. 728–737, 2010.
- [7] M. Holters and U. Zölzer, “A generalized method for the derivation of non-linear state-space models from circuit schematics,” in *23rd European Signal Process. Conf. (EUSIPCO)*, Nice, France, 2015, pp. 1078–1082.
- [8] M. Holters and U. Zölzer, “Circuit simulation with inductors and transformers based on the Jiles-Atherton model of magnetization,” in *Proc. 19th Int. Conf. Digital Audio Effects (DAFx-16)*, Brno, Czech Republic, 2016, pp. 55–60.
- [9] D. T. Yeh, J. S. Abel, and J. O. Smith, “Simplified, physically-informed models of distortion and overdrive guitar effects pedals,” in *Proc. 10th Int. Conf. Digital Audio Effects (DAFx-07)*, Bordeaux, France, 2007.
- [10] J. Macak and J. Schimmel, “Real-time guitar tube amplifier simulation using an approximation of differential equations,” in *Proc. 13th Int. Conf. Digital Audio Effects (DAFx-10)*, Graz, Austria, 2010.
- [11] J. Macak and J. Schimmel, “Real-time guitar preamp simulation using modified blockwise method and approximations,” *Eurasip J. on Advances in Signal Process.*, vol. 2011, 2011.
- [12] M. Holters and U. Zölzer, “A k-d tree based solution cache for the non-linear equation of circuit simulations,” in *24th European Signal Process. Conf. (EUSIPCO)*, Budapest, Hungary, Aug. 2016, pp. 1028–1032.
- [13] L. O. Chua and P.-M. Lin, *Computer-Aided Analysis of Electric Circuits: Algorithms and Computational Techniques*, Prentice-Hall, Englewood Cliffs, NJ, 1975.
- [14] M. Khorramizadeh and N. Mahdavi-Amiri, “An efficient algorithm for sparse null space basis problem using ABS methods,” *Numerical Algorithms*, vol. 62, no. 3, pp. 469–485, June 2012.