TOWARDS EFFICIENT EMULATION OF NONLINEAR ANALOG CIRCUITS FOR AUDIO USING CONSTRAINT STABILIZATION AND CONVEX QUADRATIC PROGRAMMING

Miguel Zea and Luis A. Rivera

Dept. of Electronics, Mechatronics and Biomedical Engineering
Universidad del Valle de Guatemala
Guatemala, GTM
{mezea, larivera}@uvg.edu.gt

ABSTRACT

This paper introduces a computationally efficient method for the emulation of nonlinear analog audio circuits by combining state-space representations, constraint stabilization, and convex quadratic programming (QP). Unlike traditional virtual analog (VA) modeling approaches or computationally demanding SPICE-based simulations, our approach reformulates the nonlinear differential-algebraic (DAE) systems that arise from analog circuit analysis into numerically stable optimization problems. The proposed method efficiently addresses the numerical challenges posed by nonlinear algebraic constraints via constraint stabilization techniques, significantly enhancing robustness and stability, suitable for real-time simulations. A canonical diode clipper circuit is presented as a test case, demonstrating that our method achieves accurate and faster emulations compared to conventional state-space methods. Furthermore, our method performs very well even at substantially lower sampling rates. Preliminary numerical experiments confirm that the proposed approach offers improved numerical stability and real-time feasibility, positioning it as a practical solution for high-fidelity audio applications.

1. INTRODUCTION

Contrary to traditional engineering disciplines, in which the embrace of digital signal processing has ushered near theory-perfect implementations of filtering and conditioning, audio signal processing seeks to emulate the richness and character of analog circuitry. This richness often arises from imperfections and nonlinearities inherent to analog components and their interactions. The prevalent approach to emulate this behavior has been through the design of virtual analog (VA) filters [1], which combine discrete-time linear system theory in the frequency domain with carefully crafted, expert-informed nonlinearities to mimic analog filter structures. The primary limitation of this approach, aside from its reliance on extensive domain-specific knowledge, is that it originates from a fundamentally linear perspective, overlooking the inherently nonlinear behavior of semiconductor devices.

An alternative approach, rooted in electrical engineering practice, is to employ SPICE-based simulations [2] for high-fidelity emulation of analog circuits. Nevertheless, this method results in large-scale nonlinear differential-algebraic systems of equations (DAE) that are computationally prohibitive for real-time implementations. Yeh et al. [3, 4] proposed a balanced strategy by representing analog circuits as differential-algebraic equations (DAE), wherein the differential components remain linear. Further advancements by Holters and Zölzer [5, 6] streamlined the derivation process and allowed for more intricate nonlinear interactions. Nonetheless, these approaches still require iterative solutions to nonlinear algebraic equations at every simulation step, severely hindering real-time applicability for complex circuits.

Machine learning-based methods, such as [7] or the ones now utilized by commercial products such as Native Instruments' Guitar Rig [8], have demonstrated promising capabilities in replicating complex nonlinear audio circuit behaviors. These methods achieve remarkable real-time efficiency and accurate emulation by leveraging trained neural networks. However, such techniques often sacrifice interpretability, depend heavily on extensive high-quality training data, and lack flexibility when applied to generalized circuit configurations.

A similar computational challenge has been addressed effectively in the simulation of rigid body dynamics, particularly in handling nonlinear algebraic constraints arising from interactions such as bodies coming into contact. Notably, Baumgarte [9] introduced a stabilization method wherein nonlinear equality constraints in DAEs are replaced with dynamic stability conditions. This transforms the constraints into a set of stabilized Ordinary Differential Equations (ODEs), simplifying numerical integration significantly.

Another successful contribution in the rigid body dynamics domain was in the convex approximation of the originally nonlinear optimization problems inherent to those systems. The initial success of the MuJoCo simulator [10], for example, was partly due to its ability to efficiently convexify nonlinear optimization problems, allowing high-fidelity simulations at low sampling rates in real time.

This work introduces modifications to Yeh's original DAE-based formulation, integrating ideas from Holters' and Zölzer's generalizations. It also draws inspiration from stabilization and optimization techniques established in rigid body dynamics simulations, particularly Baumgarte's stabilization method and convex quadratic programming formulations, to propose a numerically robust and computationally efficient method suitable for real-time nonlinear audio circuit simulation with stability guarantees.

2. MATHEMATICAL BACKGROUND

Our proposed method relies on linear control systems theory and numerical optimization. Here we present the main concepts needed.

2.1. Linear Time-Invariant (LTI) State-Space Systems

A linear time-invariant (LTI) system in state-space form is generally described by the equations:

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t), \tag{1a}$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t), \tag{1b}$$

where $\mathbf{x}(t) \in \mathbb{R}^n$ represents the state vector, $\mathbf{u}(t) \in \mathbb{R}^m$ is the input vector, and $\mathbf{y}(t) \in \mathbb{R}^p$ is the output vector [11]. $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, and $\mathbf{D}$ are constant parameter matrices that define the system's dynamics and input-output relationships. In most cases, the explicit time dependence of the quantities can be ignored for simplicity.

2.1.1. Internal and Input-Output (BIBO) Stability

An LTI system is said to be internally stable if and only if all eigenvalues of the system matrix $\mathbf{A}$ have strictly negative real parts [11], that is, $\mathrm{Re}\{\lambda_i\} < 0,\ \forall\, \lambda_i \in \sigma(\mathbf{A})$. Internal stability implies that, no matter the initial condition of the state, there is a guaranteed asymptotic and exponential convergence to the origin $\mathbf{x} = \mathbf{0}$. A system matrix whose eigenvalues satisfy this condition is referred to as Hurwitz.

Additionally, an LTI system is Bounded-Input Bounded-Output (BIBO) stable if every bounded input, $\|\mathbf{u}\| < a \in \mathbb{R}_0^+$ $\forall\, t > 0$, produces a bounded output, $\|\mathbf{y}\| < b \in \mathbb{R}_0^+\ \forall\, t > 0$. For continuous-time LTI systems, BIBO stability is ensured if the system matrix is Hurwitz. Notably, however, the converse is not true.

2.2. Baumgarte's Stabilization Technique

Baumgarte's stabilization method [9] was initially proposed for stabilizing holonomic and nonholonomic constraints in rigid body dynamics. When simulating multibody dynamics, constraints typically appear as nonlinear algebraic equations, often leading to numerical drift and instability.

Consider a general nonlinear equality constraint of the form:

$$\mathbf{g}\left(\mathbf{x}(t), t\right) = \mathbf{0}, \tag{2}$$

where $\mathbf{g} : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^m$ represents a vector of nonlinear constraints. Direct numerical integration of the DAE system derived from this constraint can accumulate numerical errors, causing constraint drift.

Baumgarte proposed augmenting the constraint equations with additional stabilization terms derived from control theory principles. Specifically, the stabilization method modifies the constraint into a second-order dynamic correction form:

$$\ddot{\mathbf{g}}\left(\mathbf{x}(t), t\right) + 2\alpha\dot{\mathbf{g}}\left(\mathbf{x}(t), t\right) + \beta^2\mathbf{g}\left(\mathbf{x}(t), t\right) = \mathbf{0}, \tag{3}$$

which can be interpreted as the linear system:

$$\begin{bmatrix}\dot{\mathbf{g}}\\ \ddot{\mathbf{g}}\end{bmatrix} = \begin{bmatrix}\mathbf{0} & \mathbf{I}\\ -\beta^2\mathbf{I} & -2\alpha\mathbf{I}\end{bmatrix}\begin{bmatrix}\mathbf{g}\\ \dot{\mathbf{g}}\end{bmatrix}, \tag{4}$$

where α and β are stabilization parameters chosen to ensure convergence and stability, i.e. to ensure that the linear system is Hurwitz. This technique transforms the original DAE into a stabilized ODE, thus enhancing numerical robustness.

2.3. Convex Quadratic Programming (QP)

Convex quadratic programming refers to the optimization problem of minimizing a convex quadratic objective function subject to linear equality and inequality constraints, defined as [12]:

$$\min_{\mathbf{x}} \quad \frac{1}{2}\mathbf{x}^\top\mathbf{Q}\mathbf{x} + \mathbf{c}^\top\mathbf{x} \tag{5a}$$

$$\text{s.t.} \quad \mathbf{A}_{\mathrm{eq}}\mathbf{x} = \mathbf{b}_{\mathrm{eq}}, \tag{5b}$$

$$\mathbf{A}_{\mathrm{ineq}}\mathbf{x} \leq \mathbf{b}_{\mathrm{ineq}}, \tag{5c}$$

where $\mathbf{Q}$ is a symmetric positive semidefinite matrix, $\mathbf{c}$ is a vector defining linear cost terms, and $\mathbf{A}_{\mathrm{eq}}$, $\mathbf{b}_{\mathrm{eq}}$, $\mathbf{A}_{\mathrm{ineq}}$, $\mathbf{b}_{\mathrm{ineq}}$ define the equality and inequality constraints, respectively.

2.3.1. Feasibility Problems

A special case of quadratic programming arises when the objective function is constant or absent. This scenario is known as a feasibility problem [12], aiming solely to determine whether a solution satisfying all given constraints exists:

$$\text{find} \quad \mathbf{x} \tag{6a}$$

$$\text{s.t.} \quad \mathbf{A}_{\mathrm{eq}}\mathbf{x} = \mathbf{b}_{\mathrm{eq}}, \tag{6b}$$

$$\mathbf{A}_{\mathrm{ineq}}\mathbf{x} \leq \mathbf{b}_{\mathrm{ineq}}. \tag{6c}$$

In the further special case when only linear equality constraints are present, the feasibility problem simplifies to solving a linear system of equations:

$$\mathbf{A}_{\mathrm{eq}}\mathbf{x} = \mathbf{b}_{\mathrm{eq}}. \tag{7}$$

This reduces the computational complexity and simplifies the solution methods.

3. PROPOSED METHOD

We start from the problem as stated by Yeh et al. in [3], in which analog circuits can be described by the DAE:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} + \mathbf{C}\mathbf{i}, \tag{8a}$$

$$\mathbf{v} = \mathbf{D}\mathbf{x} + \mathbf{E}\mathbf{u} + \mathbf{F}\mathbf{i}, \tag{8b}$$

$$\mathbf{i} = \mathbf{f}\left(\mathbf{v}\right), \tag{8c}$$

where (8a) and (8b) can be considered as a state-space LTI system with $\mathbf{u}$ and $\mathbf{i}$ as inputs. The nonlinearities are attributed to active devices, which include semiconductors and vacuum tubes. The semiconductor devices are (assumed to be) modeled as a combination of voltage-controlled nonlinear current sources (8c) representing the PN and NP junctions present in their construction, in accordance with Shockley's original approximation [13], along with passive circuit elements according to the desired model complexity. In minimal coordinates, the state vector $\mathbf{x}$ corresponds to voltages across capacitors and currents through inductors. However, deriving equations in this selection of state variables is usually difficult, as it requires a manual manipulation of quantities that yields dense matrices. The modifications proposed by Holters and Zölzer in [5] use a larger set of state variables to simplify and open up the possibility to automatize the derivation of equations

[6], drawing inspiration from sparse description methods in circuit analysis. As this work focuses on the fundamentals of the problem statement, we will not discuss the issue of selecting an appropriate coordinate system for equation derivation simplification. We will, however, use their idea to collect all the nonlinear terms $\mathbf{v}, \mathbf{i}$ in the single vector $\mathbf{q} = \begin{bmatrix}\mathbf{v}\\ \mathbf{i}\end{bmatrix} \in \mathbb{R}^r$. Under this definition, the DAE described in (8a)-(8c) changes to:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} + \mathbf{G}\mathbf{q}, \tag{9a}$$

$$\mathbf{H}\mathbf{q} = \mathbf{D}\mathbf{x} + \mathbf{E}\mathbf{u}, \tag{9b}$$

$$\mathbf{i} - \mathbf{f}(\mathbf{v}) = \mathbf{0} \triangleq \boldsymbol{\eta}(\mathbf{q}) = \mathbf{0}, \tag{9c}$$

where both (9b) and $\boldsymbol{\eta}(\mathbf{q}) = \mathbf{0}$ allow for implicit linear and nonlinear relationships between semiconductor currents and voltages, and nonlinear quantities in general. In previous works, this would be the step in which an integration scheme is proposed and implemented, followed by solving the problem as a discrete-time LTI system with a nonlinear equality constraint, implying that the nonlinear algebraic equation must be solved before stepping the dynamics forward in time. We use the ideas on constraint stabilization described in Section 2.2 to propose the following relaxation of the constraint:

$$\dot{\boldsymbol{\eta}}(\mathbf{q}) = \frac{d\boldsymbol{\eta}(\mathbf{q})}{d\mathbf{q}}\dot{\mathbf{q}} = \mathbf{K}\boldsymbol{\eta}(\mathbf{q}). \tag{10}$$

That is, we reformulate the equality constraint as a homogeneous LTI system, with $\mathbf{K}$ selected as to be Hurwitz. If this is the case then $\lim_{t\to\infty} \boldsymbol{\eta}(\mathbf{q}(t)) = \mathbf{0}$ and the constraint is exponentially asymptotically satisfied regardless of its initial value, even in the presence of numerical disturbances. A Hurwitz $\mathbf{K}$ matrix is easily achieved by selecting it as diagonal with negative coefficients. A common choice is to use the identity matrix scaled by a negative factor.

Directly solving this new problem proves to be challenging as well, given that in spite of getting rid of the nonlinear algebraic constraint in (9c), a new linear implicit one was introduced by (9b) and the dynamics of $\boldsymbol{\eta}$ in (10) are given by an implicit nonlinear ODE in $\mathbf{q}$, instead of being only dependent on $\boldsymbol{\eta}$. These particularities still need a DAE solver or at least an implicit integration scheme that allows for a mass matrix, implying that the underlying problem remains numerically problematic. It may need the inverse of likely numerically unstable matrices which would yield the problem rigid.

A more natural and numerically tractable statement for the problem can be obtained via optimization, for which algebraic constraints are part of the statement itself and the implicit relationships need not be made explicit before trying for a solution. First, we use a simple implicit integration scheme such as backward Euler to discretize, but not solve for, the LTI state equation (9a):

$$(\mathbf{I} - \mathbf{A}\Delta t)\,\mathbf{x}_k = \mathbf{x}_{k-1} + (\mathbf{B}\Delta t)\mathbf{u}_k + (\mathbf{G}\Delta t)\mathbf{q}_k, \tag{11}$$

where $\Delta t = 1/f_s$ is the sampling period and the subscript k denotes the value of quantities at time step $k\Delta t$.

Next, we use an explicit scheme to discretize (10) to associate the nonlinearities only with $\mathbf{q}_{k-1}$ instead of $\mathbf{q}_k$:

$$\frac{d\boldsymbol{\eta}(\mathbf{q}_{k-1})}{d\mathbf{q}}(\mathbf{q}_k - \mathbf{q}_{k-1}) = (\mathbf{K}\Delta t)\boldsymbol{\eta}(\mathbf{q}_{k-1}). \tag{12}$$

With this, we are then able to define the following convex QP using $\mathbf{x}_k$ and $\mathbf{q}_k$ as decision variables:

$$\begin{bmatrix}\mathbf{x}_k\\ \mathbf{q}_k\end{bmatrix} = \arg\min_{\mathbf{x}_k,\mathbf{q}_k} \quad \frac{1}{2}\begin{bmatrix}\mathbf{x}_k^\top & \mathbf{q}_k^\top\end{bmatrix}\mathbf{Q}\begin{bmatrix}\mathbf{x}_k\\ \mathbf{q}_k\end{bmatrix} + \mathbf{c}^T\begin{bmatrix}\mathbf{x}_k\\ \mathbf{q}_k\end{bmatrix} \tag{13a}$$

$$\text{s.t.} \quad \begin{bmatrix}\mathbf{I} - \mathbf{A}\Delta t & -\mathbf{G}\Delta t\end{bmatrix}\begin{bmatrix}\mathbf{x}_k\\ \mathbf{q}_k\end{bmatrix} = \mathbf{x}_{k-1} + (\mathbf{B}\Delta t)\mathbf{u}_k, \tag{13b}$$

$$\begin{bmatrix}-\mathbf{D} & \mathbf{H}\end{bmatrix}\begin{bmatrix}\mathbf{x}_k\\ \mathbf{q}_k\end{bmatrix} = \mathbf{E}\mathbf{u}_k, \tag{13c}$$

$$\begin{bmatrix}\mathbf{0} & \frac{d\boldsymbol{\eta}(\mathbf{q}_{k-1})}{d\mathbf{q}}\end{bmatrix}\begin{bmatrix}\mathbf{x}_k\\ \mathbf{q}_k\end{bmatrix} = (\mathbf{K}\Delta t)\boldsymbol{\eta}(\mathbf{q}_{k-1}) + \frac{d\boldsymbol{\eta}(\mathbf{q}_{k-1})}{d\mathbf{q}}\mathbf{q}_{k-1}, \tag{13d}$$

$$\mathbf{A}_{\text{ineq}}\begin{bmatrix}\mathbf{x}_k\\ \mathbf{q}_k\end{bmatrix} \leq \mathbf{b}_{\text{ineq}}. \tag{13e}$$

$\mathbf{Q}$ and $\mathbf{c}$ are the constant coefficient matrix and vector, respectively. Alternatively, we can define a new set of decision variables as $\mathbf{z}_k = \begin{bmatrix}\mathbf{x}_k\\ \mathbf{q}_k\end{bmatrix}$, and obtain:

$$\mathbf{z}_k = \arg\min_{\mathbf{z}_k} \quad \frac{1}{2}\mathbf{z}_k^\top\mathbf{Q}\mathbf{z} + \mathbf{c}^\top\mathbf{z}_k \tag{14a}$$

$$\text{s.t.} \quad \mathbf{A}_{\text{eq}}\mathbf{z}_k = \mathbf{b}_{\text{eq}}, \tag{14b}$$

$$\mathbf{A}_{\text{ineq}}\mathbf{z}_k \leq \mathbf{b}_{\text{ineq}}, \tag{14c}$$

where

$$\mathbf{A}_{\text{eq}} = \begin{bmatrix}\mathbf{I} - \mathbf{A}\Delta t & -\mathbf{G}\Delta t\\ -\mathbf{D} & \mathbf{H}\\ \mathbf{0} & \frac{d\boldsymbol{\eta}(\mathbf{q}_{k-1})}{d\mathbf{q}}\end{bmatrix}, \tag{15}$$

$$\mathbf{b}_{\text{eq}} = \begin{bmatrix}\mathbf{x}_{k-1} + (\mathbf{B}\Delta t)\mathbf{u}_k\\ \mathbf{E}\mathbf{u}_k\\ (\mathbf{K}\Delta t)\boldsymbol{\eta}(\mathbf{q}_{k-1}) + \frac{d\boldsymbol{\eta}(\mathbf{q}_{k-1})}{d\mathbf{q}}\mathbf{q}_{k-1}\end{bmatrix}. \tag{16}$$

This QP's solution not only satisfies the constraints imposed by the original DAE problem but allows for additional flexibility and safety in the form of the cost function (14a), and linear inequality constraints (14c). For example, (14c) can be used to bound $\mathbf{q}$ as to ensure that (9a) is BIBO stable, as the (user) input is assumed to be bounded as well. This can also be done via (14a), in addition to penalizing state divergence.

In the simplest of cases, for which there are no restrictions on cost or bounds, the QP reduces to a feasibility problem with only linear equality constraints which, in turn, reduces to the solution of the linear system (14b) in each time step. These conditions make this the competing method to the ones proposed in [3], [5] and [14], and the fact that it is posed as a linear system avoids the need of using matrix inversion and allows leveraging efficient linear solvers, both dense and sparse.

Finally, required circuit outputs can be obtained using:

$$\mathbf{y}_k = \mathbf{M}\mathbf{x}_k + \mathbf{L}\mathbf{u}_k + \mathbf{N}\mathbf{q}_k, \tag{17}$$

as similarly stated in [3] and [4].

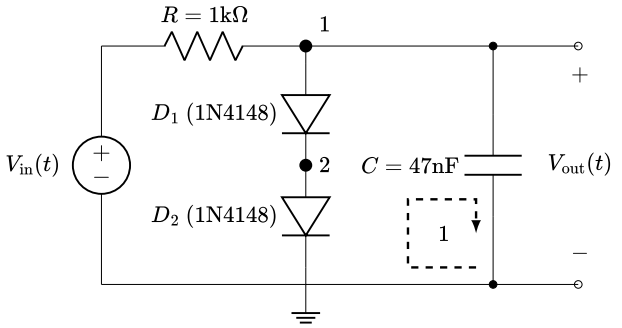

Figure 1: *Schematic of the simple diode clipper used as example, as proposed in [5].*

4. A CANONICAL EXAMPLE

To test the proposed QP method, we reexamined the simple diode clipper of Figure 1, used as an example for the state-space method in [5]. Before presenting the specifics of the application of our proposed method, for comparison purposes, it is worth explicitly describing the derivation and simulation process of the state-space method. The system equations are given by:

$$\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1} + \mathbf{B}\mathbf{u}_k + \mathbf{C}\mathbf{z}_k, \tag{18a}$$

$$\mathbf{y}_k = \mathbf{D}\mathbf{x}_{k-1} + \mathbf{E}\mathbf{u}_k + \mathbf{F}\mathbf{z}_k, \tag{18b}$$

$$\mathbf{q}_k = \mathbf{D}_q\mathbf{x}_{k-1} + \mathbf{E}_q\mathbf{u}_k + \mathbf{F}_q\mathbf{z}_k, \tag{18c}$$

$$\mathbf{f}\left(\mathbf{q}_k\right) = \mathbf{0}. \tag{18d}$$

Note that the terms in both methods need not be the same. For this particular example, we use those found in [5]:

$$\mathbf{A} = -1, \quad \mathbf{B} = 0, \quad \mathbf{C} = \begin{bmatrix} 94 \times 10^{-9} & 0 \end{bmatrix}, \tag{19a}$$

$$\mathbf{D} = 0, \quad \mathbf{E} = 0, \quad \mathbf{F} = \begin{bmatrix} 1 & 0 \end{bmatrix}, \tag{19b}$$

$$\mathbf{D}_q = \begin{bmatrix} 0 \\ 88200 \\ 0 \\ 88200 \end{bmatrix}, \quad \mathbf{E}_q = \begin{bmatrix} 0 \\ 1 \times 10^{-3} \\ 0 \\ 1 \times 10^{-3} \end{bmatrix}, \tag{19c}$$

$$\mathbf{f}(\mathbf{q}) = \begin{bmatrix} I_s\left(e^{q_1/(nV_T)} - 1\right) - q_2 \\ I_s\left(e^{q_3/(nV_T)} - 1\right) - q_4 \end{bmatrix}. \tag{19d}$$

In both the state-space method and our method, it is assumed that the diodes are modeled by Shockley's equation [13]. For the 1N4148 diode, values of $I_s = 2.9 \times 10^{-9}$ and $n = 1.75 \times 10^{-3}$ can be extracted from the data-sheet's voltage-current curves [15], assuming a typical value of $V_T = 25.85$ mV.

The implementation of the state-space method uses the previous state and present input to solve the nonlinear algebraic system (18c)-(18d) for $\mathbf{z}_k$. Then, $\mathbf{z}_k$ is used to propagate (18a)-(18c) forward one step in time, and the process repeats iteratively.

For the proposed QP method, we apply Kirchhoff's Voltage Law in loop 1 and Kirchhoff's Current Law in nodes 1-2 of the circuit in Figure 1. This yields the equations:

$$V_{D_1} + V_{D_2} - V_C = 0, \tag{20a}$$

$$\frac{V_{\text{in}} - V_C}{R} - I_{D_1} - C\dot{V}_C = 0, \tag{20b}$$

$$I_{D_1} - I_{D_2} = 0, \tag{20c}$$

which can then be used to obtain the system's matrices as

$$\mathbf{A} = -\frac{1}{RC}, \quad \mathbf{B} = \frac{1}{RC}, \quad \mathbf{G} = \begin{bmatrix} 0 & 0 & -1/C & 0 \end{bmatrix}, \tag{21a}$$

$$\mathbf{H} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix}, \quad \mathbf{D} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \mathbf{E} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \tag{21b}$$

$$\mathbf{M} = 1, \quad \mathbf{L} = 0, \quad \mathbf{N} = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}, \tag{21c}$$

$$\boldsymbol{\eta}(\mathbf{q}) = \begin{bmatrix} q_3 - I_s\left(e^{q_1/(nV_T)} - 1\right) \\ q_4 - I_s\left(e^{q_2/(nV_T)} - 1\right) \end{bmatrix}, \quad \mathbf{K} = -(1/\Delta t)\mathbf{I}, \tag{21d}$$

$$\frac{d\boldsymbol{\eta}\left(\mathbf{q}\right)}{d\mathbf{q}} = \frac{I_s}{nV_T}\begin{bmatrix} -e^{q_1/(nV_T)} & 0 & \frac{nV_T}{I_s} & 0 \\ 0 & -e^{q_2/(nV_T)} & 0 & \frac{nV_T}{I_s} \end{bmatrix}, \tag{21e}$$

under the following definitions:

$$x = V_C, \quad u = V_{\text{in}}, \quad y = V_{\text{out}} = V_C, \tag{22a}$$

$$q_1 = V_{D_1}, \quad q_2 = V_{D_2}, \quad q_3 = I_{D_1}, \quad q_4 = I_{D_2}. \tag{22b}$$

These matrices are then used along with $\mathbf{Q} = \mathbf{I}$, $\mathbf{c} = \mathbf{0}$, and $\mathbf{A}_{\text{ineq}} = \mathbf{0}$ in (14a)-(14c) to iteratively solve for the state and nonlinear quantities of the clipper.

The QP method finds the same solution as the state-space method, as shown in Figure 2. However, the zoomed-in comparison presented in Figure 3 shows that the QP method adheres more closely to the results obtained with SPICE. It is important to mention that, for both the state-space and QP methods, we used the same maximum number of iterations and error threshold for the numerical solvers, in order to have a fair comparison between their performance. Given this, the differences in the outputs could be explained by the modification in (10), which acts as a stabilizing term for the nonlinear algebraic constraint, and thus, the QP method is only subject to the numerical disturbances associated with the integration scheme.

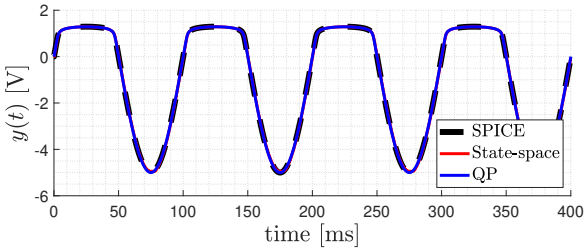

Figure 2: *Comparison of the clipper's output for SPICE, state-space and QP methods for a 5 V, 10 Hz sine wave, using $f_s = 44.1$ kHz and zero initial conditions.*

To quantitatively measure the observed differences in Figure 3, we calculated the squared error between the output obtained via SPICE and the results of both state-space and QP methods. This is shown in Figure 4, where the QP method consistently outperformed the state-space method in terms of approximation noise. Note that in both cases this noise is likely to be inaudible. Figure 5 shows that this behavior extends to lower sampler rates, for which the QP still performs adequately even after a down-sampling factor of 128, thus presenting a strong case for its use in real-time applications. Figure 6 shows how down-sampling can affect the output of the state-space method, which involves both magnitude

and phase. The output of the QP method, on the other hand, remains practically unaffected.

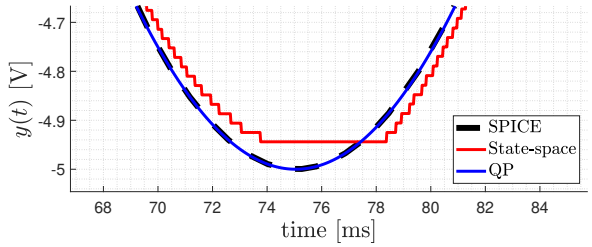

Figure 3: *Zoomed-in comparison of the clipper's output for SPICE, state-space and QP methods for a 5 V, 10 Hz sine wave, using $f_s = 44.1$ kHz and zero initial conditions.*

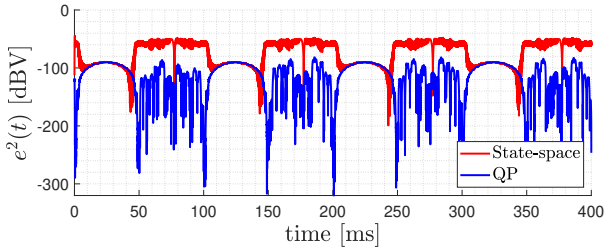

Figure 4: *Comparison of the squared error between SPICE and the state-space method and between SPICE and the QP method, using $f_s = 44.1$ kHz.*

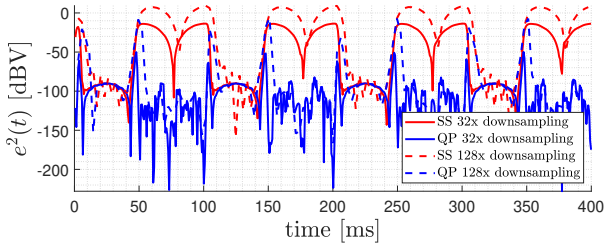

Figure 5: *Comparison of the squared error between SPICE and the state-space (SS) method and between SPICE and the QP method, using $f_s = 44.1/32$ and $f_s = 44.1/128$ kHz.*

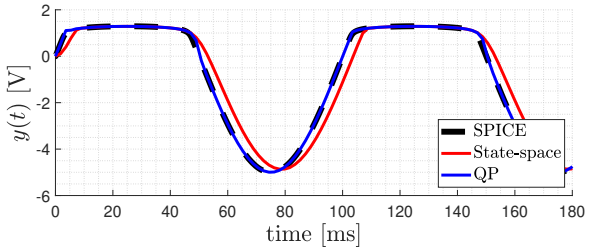

Figure 6: *Comparison of the clipper's output for SPICE, state-space and QP methods for a 5 V, 10 Hz sine wave, using $f_s = 44.1/80$ kHz and zero initial conditions.*

Preliminary testing in MATLAB showed that the convergence time of both methods was in the same order of magnitude. The performance of the QP method can be boosted by turning the QP into a feasibility problem with only linear equality constraints by using $\mathbf{Q} = \mathbf{0}$ in addition to $\mathbf{A}_{\text{ineq}} = \mathbf{0}$. This is equivalent to only solving the linear problem (14b) for each iteration, and it allows for a convergence time of an order of magnitude less than the state-space method.

Finally, we conducted additional experiments using input waves at different frequencies. Figures 7 - 8 show the clipper's output for both methods with inputs at 1 and 100 Hz, respectively. It is important to notice that, qualitatively, both methods still follow the waveform obtained by SPICE. However, the error plots show that the QP method peforms better, as before.

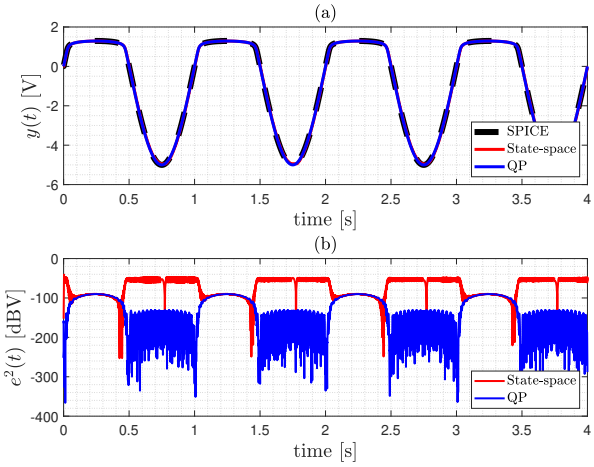

Figure 7: *(a) Comparison of the clipper's output for SPICE, state-space and QP methods for a 5 V, 1 Hz sine wave, using $f_s = 44.1$ kHz and zero initial conditions. (b) Comparison of the squared error between SPICE and the state-space method and between SPICE and the QP method, for the same conditions as in (a).*

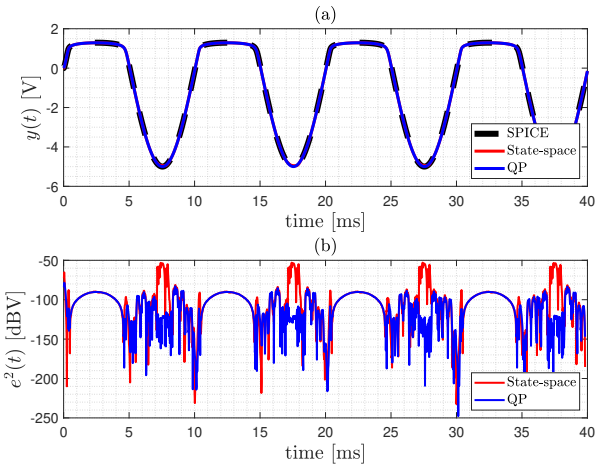

Figure 8: *(a) Comparison of the clipper's output for SPICE, state-space and QP methods for a 5 V, 100 Hz sine wave, using $f_s = 44.1$ kHz and zero initial conditions. (b) Comparison of the squared error between SPICE and the state-space method and between SPICE and the QP method, for the same conditions as in (a).*

5. CONCLUSIONS AND FUTURE WORK

In this work, we presented an innovative approach for emulating nonlinear analog audio circuits by integrating constraint stabilization and convex quadratic programming into state-space models. Our formulation successfully transforms the computational challenge of nonlinear algebraic constraints into a numerically stable optimization framework, providing significant advantages in terms of robustness and computational efficiency.

Through validation with a canonical diode clipper circuit, the method demonstrated accuracy comparable to conventional methods while showing superior stability and performance even at reduced sampling rates. The reduction in computational demands underscores the potential applicability in real-time digital audio effects, where maintaining audio fidelity without prohibitive computational costs is crucial.

Future work includes extending this method to more complex analog audio circuits and investigating adaptive schemes for automatic tuning of stabilization parameters, further enhancing numerical performance. Furthermore, if sparse methods are employed for deriving the circuit's equations, the QP formulation becomes amenable to using Krylov subspace methods for efficiently solving the feasibility problem, or fast sparse QP solvers such as [16] for the full optimization problem. Additionally, exploring the possibility of dynamically changing circuit component values within iterations without significantly impacting performance represents another promising direction.

6. ACKNOWLEDGMENTS

We would like to thank Pablo Oliva, Diego Morales, Jacqueline Guarcax, Doménico Dellachiessa and José Pablo Marroquín, for their contributions in the preliminary stages of this work. We would also like to thank Carlos Esquit and the Department of Electronics, Mechatronics and Biomedical Engineering at UVG, for the financial support during the development of this work.

7. REFERENCES

[1] V. Zavalishin, "The Art of VA Filter Design," Available at https://www.native-instruments.com/fileadmin/ni_media/ downloads/pdf/ VAFilterDesign_2.1.0.pdf, accessed March 25, 2025.

[2] A. Vladimirescu, *The spice book*, J. Wiley, New York, NY, USA, 1994.

[3] 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 Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 4, pp. 728–737, 2010.

[4] D. T. Yeh, "Automated physical modeling of nonlinear audio circuits for real-time audio effects—part II: BJT and vacuum tube examples," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 4, pp. 1207–1216, 2012.

[5] M. Holters and U. Zölzer, "A generalized method for the derivation of non-linear state-space models from circuit schematics," in *2015 23rd European Signal Processing Conference (EUSIPCO)*, Nice, France, Aug.-Sept. 2015, pp. 1073–1077.

[6] M. Holters and U. Zölzer, "Automatic decomposition of nonlinear equation systems in audio effect circuit simulation," in *Proc. of the 20th Int. Conference on Digital Audio Effects (DAFx-17)*, Edinburgh, UK, Sept. 2017, pp. 138–144.

[7] Š. Miklánek, A. Wright, V. Välimäki, and J. Schimmel, "Neural grey-box guitar amplifier modelling with limited data," in *Proc. of the 26th Int. Conference on Digital Audio Effects (DAFx-23)*, Copenhagen, Denmark, Sept. 2023, pp. 1–7.

[8] J. D. Parker, F. Esqueda, and A. Bergner, "Modelling of nonlinear state-space systems using a deep neural network," in *Proc. of the 22nd Int. Conference on Digital Audio Effects (DAFx-19)*, Birmingham, UK, Sept. 2019, pp. 1–8.

[9] J. Baumgarte, "Stabilization of constraints and integrals of motion in dynamical systems," *Computer Methods in Applied Mechanics and Engineering*, vol. 1, no. 1, pp. 1–16, 1972.

[10] Emanuel T., "Convex and analytically-invertible dynamics with contacts and constraints: Theory and implementation in MuJoCo," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, Hong Kong, China, May-Jun. 2014, pp. 6054–6061.

[11] C. Chen, *Linear System Theory and design*, Oxford University Press, New York, NY, USA, 2013.

[12] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, Cambridge, UK, 2004.

[13] W. Shockley, "The theory of p-n junctions in semiconductors and p-n junction transistors," *Bell System Technical Journal*, vol. 28, no. 3, pp. 435–489, 1949.

[14] B. Holmes and M. van Walstijn, "Improving the robustness of the iterative solver in state-space modelling of guitar distortion circuitry," in *Proc. of the 18th Int. Conference on Digital Audio Effects (DAFx-15)*, Trondheim, Norway, Nov.-Dec. 2015, pp. 1–8.

[15] "1N4148; 1N4448 high-speed diodes datasheet," Tech. Rep., NXP Semiconductors, 2004.

[16] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd, "OSQP: an operator splitting solver for quadratic programs," *Mathematical Programming Computation*, vol. 12, no. 4, pp. 637–672, 2020.