

MODELING NONLINEAR CIRCUITS WITH LINEARIZED DYNAMICAL MODELS VIA KERNEL REGRESSION

Daniel J. Gillespie

Columbia University / Eventide Inc.
2960 Broadway, New York, NY 10027 /
1 Alsan Way, Little Ferry, NJ 07643
djg2138@columbia.edu / DGillespie@eventide.com

Daniel P. W. Ellis

Columbia University
2960 Broadway, New York, NY 10027
dpwe@ee.columbia.edu

ABSTRACT

This paper introduces a novel method for the solution of guitar distortion circuits based on the use of kernels. The proposed algorithm uses a kernel regression framework to linearize the inherent nonlinear dynamical systems created by such circuits and proposes data and kernel selection algorithms well suited to learn the required regression parameters. Examples are presented using the One Capacitor Diode Clipper and the Common-Cathode Tube Amplifier.

Index Terms— Circuit Modeling, Nonlinear Dynamical Models, Kernel Regression, Linearization

1. INTRODUCTION

The accurate digital modeling of guitar distortion circuits has been an interest for several decades, with solutions proposed in both the research literature and the commercial space[1][2][3][4]. Elegant solutions to this problem have remained elusive mainly because common distortion circuits form highly nonlinear dynamical systems which have no closed form solutions. Because of this, existing solutions consist of either numerical methods which don't run in a fixed time[2][4], or relaxations of the nonlinear dynamical system for which a closed form solution does exist, such as separation of the nonlinearity from the dynamical system[1]. Alternatively, in this paper we will use kernel methods to propose a linearization of the dynamical system which will allow for a closed form solution and an algorithm which models the circuit in fixed time using a support vector machine.

Section 2 will review the K-Method[2], an existing nonlinear dynamical system circuit model which will form the basis for our derivation. Section 3 will describe the linearization of the dynamical system via an injection into a high dimensional space, and the derivation of a closed form solution. Section 4 will describe a solution to the problem via kernel regression using a Support Vector Machine. Section 5 will introduce methods for finding the proper injection map Φ which performs the linearization. Section 6 will describe our experiments and results. Finally, Section 7 will present our conclusions.

2. CIRCUITS AS NONLINEAR DYNAMICAL SYSTEMS

There are several formulations of a circuit as a nonlinear dynamical system. In this paper we will use the K-Method as our basis

formulation[2], which is based on the Kirchhoff variables representing Voltage and Current. This defines the dynamical system as an ordinary differential equation (ODE) of the form

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} + \mathbf{C}\mathbf{i} \quad (1)$$

$$\mathbf{i} = f(\mathbf{D}\mathbf{x} + \mathbf{E}\mathbf{u} + \mathbf{F}\mathbf{i}) \quad (2)$$

$$y = \mathbf{L}\mathbf{x} + \mathbf{M}\mathbf{u} + \mathbf{N}\mathbf{i} \quad (3)$$

where the vector \mathbf{u} represents the inputs to the circuit, \mathbf{x} represents the states of the circuit and $\dot{\mathbf{x}}$ their first derivatives with respect to time. In the K-Method, the state is defined as the voltages across the capacitors and the currents through the inductors. The scalar value y represents the output of the circuit. The vector \mathbf{i} represents the output currents of the nonlinear elements in the circuit, which in guitar distortion circuits are often based on the control voltages for the diodes, transistors and vacuum tubes in the circuit. The nonlinearity is captured by the function $f(\cdot)$. And finally, the matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}, \mathbf{E}, \mathbf{F}, \mathbf{L}, \mathbf{M}$, and \mathbf{N} are the linear mixing parameters defined by the connections and values of the passive elements in the circuit.

3. LINEARIZATION AND SOLUTION OF A NONLINEAR DYNAMICAL SYSTEM

If we define the vector $\mathbf{z} := \{\mathbf{x}, \mathbf{u}\}$, of length N , we can rewrite the system of (1), (2), (3) as

$$\dot{\mathbf{z}} = \hat{\mathbf{H}}\mathbf{z} + \hat{\mathbf{C}}\mathbf{i} \quad (4)$$

$$\mathbf{i} = f(\mathbf{J}\mathbf{z} + \mathbf{F}\mathbf{i}) \quad (5)$$

$$y = \mathbf{K}\mathbf{z} + \mathbf{N}\mathbf{i} \quad (6)$$

where $\hat{\mathbf{H}} := \{\mathbf{A}, \mathbf{B}\}$ and an unknown component for the calculation of $\dot{\mathbf{u}}$, $\hat{\mathbf{C}}$ is \mathbf{C} plus a component for $\dot{\mathbf{u}}$, $\mathbf{J} := \{\mathbf{D}, \mathbf{E}\}$, and $\mathbf{K} := \{\mathbf{L}, \mathbf{M}\}$. Note that this adds an undefined set of constraints on $\dot{\mathbf{u}}$, but it doesn't matter because we will learn them later.

Because eqn.(4) and eqn.(6) are both nonlinear functions of \mathbf{z} only, we can rewrite the system as pair of nonlinear functions, $g(\cdot)$ and $h(\cdot)$

$$\dot{\mathbf{z}} = g(\mathbf{z}) \quad (7)$$

$$y = h(\mathbf{z}) \quad (8)$$

which replace the coefficient matrices and the nonlinearity $f(\cdot)$.

Now we will introduce a map $\Phi^y(\cdot) : R^N \rightarrow Y$ which forms an injection into a high dimensional space Y where the nonlinear function $g(\cdot)$ can be replaced by the linear operator \mathbf{G}^ϕ , and the nonlinear function $h(\cdot)$ can be replaced by the linear operator \mathbf{H}^ϕ [5].

This work was supported by the NSF grant IIS-1117015.

$$\Phi^y(\cdot) : R^N \rightarrow Y; s.t. : g(\mathbf{x}) = \mathbf{G}^{\phi T} \Phi^y(\mathbf{x}), h(\mathbf{x}) = \mathbf{H}^{\phi T} \Phi^y(\mathbf{x}) \quad (9)$$

We further require that Φ^y has an associated kernel function $k_\phi(\cdot, \cdot)$ such that $k_\phi(x, y) = \langle \Phi^y(\mathbf{x}), \Phi^y(\mathbf{y}) \rangle$, the inner product between the two vectors in high dimensional space.

Now, injecting \mathbf{z} into Y allows us to linearize the dynamical system by rewriting (7) and (8) as

$$\dot{\mathbf{z}}^\phi = \mathbf{G}^\phi \mathbf{z}^\phi \quad (10)$$

$$y = \mathbf{H}^\phi \mathbf{z}^\phi \quad (11)$$

for the proper choices of \mathbf{G}^ϕ , \mathbf{H}^ϕ , and $\mathbf{z}^\phi = \Phi^y(\mathbf{z})$, the injection of the state.

We can now discretize the system using the reverse Euler integration rule,

$$\dot{\mathbf{z}}_n^\phi = \alpha(\mathbf{z}_n^\phi - \mathbf{z}_{n-1}^\phi) \quad (12)$$

This allows us to derive a discrete linear dynamical system in Y ,

$$\mathbf{z}_n^\phi = \alpha(\alpha \mathbf{I} - \mathbf{G})^{-1} \mathbf{z}_{n-1}^\phi \quad (13)$$

$$y_n = \mathbf{H}^\phi \mathbf{z}_n^\phi \quad (14)$$

or simply as the one step update rule

$$y_n = \alpha \mathbf{H}^\phi (\alpha \mathbf{I} - \mathbf{G}^\phi)^{-1} \mathbf{z}_{n-1}^\phi \quad (15)$$

4. SOLUTION VIA KERNEL REGRESSION

Now, using a change of variables we can rewrite equation (15) as a linear regression problem in Y .

$$y_n = \beta \hat{\mathbf{Z}}^{\phi T} \mathbf{z}_{n-1}^\phi \quad (16)$$

where $\hat{\mathbf{Z}}^{\phi}$ is a matrix whose rows are training examples of \mathbf{z}^ϕ and β is a set of weights on those training examples learned by a support vector regressor[6]. In fact this can be rearranged to write the classic SVR formulation

$$y_n = \sum_{i=1}^l (\beta_i - \beta_i^*) \langle \mathbf{z}_i^\phi, \mathbf{z}_{n-1}^\phi \rangle \quad (17)$$

where the i^{th} element of β is defined as $(\beta_i - \beta_i^*)$, the i^{th} row of $\hat{\mathbf{Z}}^{\phi T}$ is the training point \mathbf{z}_i^ϕ in Y , the offset (commonly b) is integrated into the dimension of \mathbf{z}^ϕ , and l is the total number of training points. By using our kernel function $k_\phi(\cdot, \cdot) = \langle \cdot, \cdot \rangle_Y$ we can now write our solution for y_n solely in terms of the input variables \mathbf{z}

$$y_n = \sum_{i=1}^l (\beta_i - \beta_i^*) k_Y(\mathbf{z}_i, \mathbf{z}_{n-1}) \quad (18)$$

We can use a similar procedure to find the unknown x_n^d representing each dimension $d \in D$ of \mathbf{x}_n . To do this, we define a map from the original state space of \mathbf{z} into a new high dimensional space X^d , $\Psi^{x^d}(\cdot) : R^N \rightarrow X^d$, and a weight vector ρ^d over training points $\hat{\mathbf{Z}}^{\psi^d}$ to learn

$$x_n^d = \rho^d \hat{\mathbf{Z}}^{\psi^d T} \mathbf{z}_{n-1}^{\psi^d} \quad (19)$$

or

$$x_n^d = \sum_{i=1}^l (\rho_i^d - \rho_i^{d*}) k_{X^d}(\mathbf{z}_i^d, \mathbf{z}_{n-1}) \quad (20)$$

where \mathbf{z}_i^d is the i^{th} training point, stored in the i^{th} row of the training data matrix \mathbf{Z}^d

This will allow us to solve our dynamical equation by running D update equations of the form (20) and one solution equation of the form (18) for each point n .

5. FINDING Φ

This linearization of the dynamical system is dependent on our ability to find a set of maps $\Phi^y(\cdot)$, $\Psi^{x^d}(\cdot)$ under which the nonlinear functions $g(\cdot)$, $h(\cdot)$ can be represented by the linear operators \mathbf{G} , \mathbf{H} . While this concept has solid theoretical support and has been developed for certain categories of kernels[5], the selection of the map and its associated kernel function is potentially very difficult and its existence is not guaranteed.

In common practice, kernels with a low number of parameters are used, and kernel selection is achieved by a grid search over the kernel parameters. However, there are several existing kernel selection methods which attempt to use a more intelligent algorithm to make an optimal kernel choice. Two of these methods include gradient descent over the kernel parameters[7] and learning the weights on a linear combination of basis kernels[8]. In this work we will experiment with both methods. First we try gradient descent on the parameters of the Mahalanobis Kernel[9], which is a generalization of the popular Gaussian RBF kernel, whose individual dimensions each have their own kernel parameter, γ_i . It is defined by

$$k(\mathbf{x}, \mathbf{y}) = \exp(-\sum_i \gamma_i (x_i - y_i)^2) \quad (21)$$

where x_i, y_i is the i^{th} dimension of \mathbf{x}, \mathbf{y} and γ_i is the i^{th} element of the kernel parameter γ .

Secondly, on the intuition that the nonlinear circuits that we are dealing with are essentially distortions of the inherently linear dynamical systems formed by linear circuits, we will form a kernel which is a linear combination of a linear kernel and a gaussian RBF. This creates the kernel equation

$$k(\mathbf{x}, \mathbf{y}) = \alpha \mathbf{x}^T \mathbf{y} + (1 - \alpha) \exp(-\gamma \|\mathbf{x} - \mathbf{y}\|^2) \quad (22)$$

with kernel parameters α and γ .

Furthermore, instead of finding a set of maps which form a \mathbf{G}, \mathbf{H} that precisely linearizes $g(\mathbf{x}^\phi), h(\mathbf{x}^\phi); \forall \mathbf{x}^\phi \in Y$, we will relax the problem to one of finding a set of maps which form a \mathbf{G}, \mathbf{H} which closely approximates $g(\cdot), h(\cdot)$ over the domain of \mathbf{z} . To do so, we need a measurement to describe how close our approximation is, an estimate for the domain of \mathbf{z} , and the best choice for the kernel parameters which create our map. All three of these issues are addressed by our proposed Add The Worst algorithm.

The proposed algorithm, described in Algorithm 1, initializes a working set with the extrema of the training data in order to reliably cover the system domain. Then it repeatedly finds support vector regressors for the output y_n and current state \mathbf{x}_n , finds the testing points with the greatest prediction error, and moves these points from the testing set to the working set. It also re-estimates the idea kernel parameters in each step. This results in a kernel map and associated training data set which form operators \mathbf{G}, \mathbf{H} which closely approximate $g(\cdot), h(\cdot)$ over a reasonable estimate of the domain of \mathbf{z} .

Data: $\mathbf{u}, \mathbf{x}, \mathbf{y}$ vectors for a nonlinear circuit

Result: $\beta, \rho^d, \hat{\mathbf{Z}}, \hat{\mathbf{Z}}^d$ parameters for circuit synthesis

1. Form $\mathbf{Z}_t, \mathbf{Z}_t^d$ training arrays from \mathbf{u}, \mathbf{x} vectors and label vectors $\mathbf{y}_n, \mathbf{x}_n^d$;

2. Find the 2D points which represent the extrema of $\mathbf{Z}_t, \mathbf{Z}_t^d$;

3. Initialize $\hat{\mathbf{Z}}, \hat{\mathbf{Z}}^d$ by moving these data points from $\mathbf{Z}_t, \mathbf{Z}_t^d, \mathbf{y}_n, \mathbf{x}_n^d$ to $\hat{\mathbf{Z}}, \hat{\mathbf{Z}}^d, \hat{\mathbf{y}}_n, \hat{\mathbf{x}}_n^d$;

while $\text{length}(\hat{\mathbf{Z}}, \hat{\mathbf{Z}}^d) < \text{targetLength}$ **do**

4. Set γ, γ^d by gradient descent over $\hat{\mathbf{Z}}, \hat{\mathbf{Z}}^d$;

5. Train a set of SVM's with $\hat{\mathbf{Z}}, \hat{\mathbf{Z}}^d, \gamma, \gamma^d, \hat{\mathbf{y}}_n, \hat{\mathbf{x}}_n^d$ to determine β, ρ^d ;

6. Use models $\hat{\mathbf{Z}}, \hat{\mathbf{Z}}^d, \beta, \rho^d$ on data $\mathbf{Z}_t, \mathbf{Z}_t^d$ to build label estimates $\tilde{\mathbf{y}}_n, \tilde{\mathbf{x}}_n^d$;

7. Find errors, $\epsilon_{y_n} = |\mathbf{y}_n - \tilde{\mathbf{y}}_n|^2, \epsilon_{x_n^d} = |\mathbf{x}_n^d - \tilde{\mathbf{x}}_n^d|^2$;

8. Find coordinates of $\max \epsilon_{y_n}, \max \epsilon_{x_n^d}$ and move their associated points from $\mathbf{Z}_t, \mathbf{Z}_t^d, \mathbf{y}_n, \mathbf{x}_n^d$ to

$\hat{\mathbf{Z}}, \hat{\mathbf{Z}}^d, \hat{\mathbf{y}}_n, \hat{\mathbf{x}}_n^d$;

end

Algorithm 1: Add The Worst Algorithm for Training Data and Kernel Selection

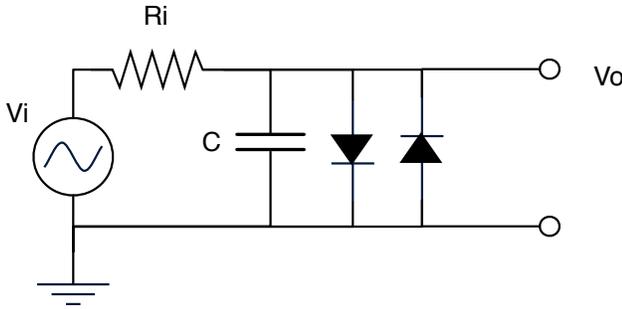


Figure 1: Schematic of a One Capacitor Diode Clipper.

6. EXPERIMENTS

To determine the success of our algorithm, we tested the proposed method using two common nonlinear guitar sub-circuits, the One-Capacitor Diode Clipper, and the Common-Cathode Tube Amplifier.

6.1. One Capacitor Diode Clipper

The One-Capacitor Diode Clipper is shown in Fig.1. It is an essential nonlinear audio circuit used in everything from guitar distortion pedals[10] to analog synthesizer filters[11]. It is essentially an RC lowpass filter with a pair of diodes across the capacitor. The diodes limit the output voltage swing to the respective diode's turn on voltage by shunting current around the capacitor whenever either diode is conducting. Because diodes smoothly lower their resistance in proportion to the voltage across them, the result is a soft saturation which is considered sonically pleasing.

The One-Capacitor Diode Clipper has one state dimension which is the voltage across the capacitor C . When discretized,

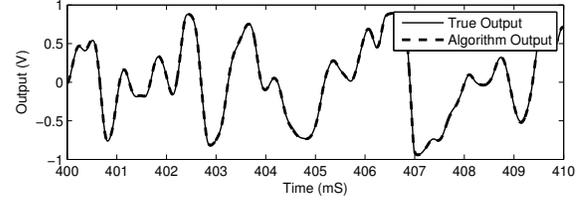


Figure 2: Comparison of our Algorithm to LTSpice Simulation of a One Capacitor Diode Clipper with Guitar Input.

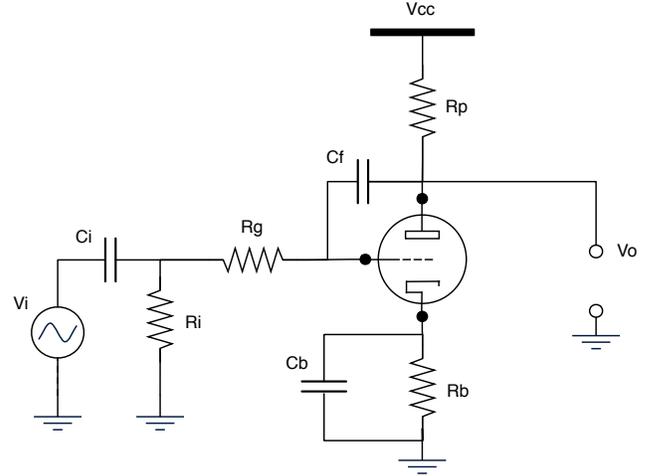


Figure 3: Schematic of a Common-Cathode Tube Amplifier.

it is equivalent to the previous output y_{n-1} . For the purposes of this experiment we synthesized the circuit using Linear Technologies LTSpiceIV's model of a 1N4001 signal diode, and with passive component values of $R_i = 2k\Omega$ and $C = 0.047\mu F$. We used the Add The Worst algorithm with 5 seconds of recorded guitar to train a 50, 100, 300, and 500 point model, using the linear-rbf kernel from (22). In order to reduce the likelihood of errors introduced by digital aliasing, we used a sample rate of 384kHz for all tests. We then tested the models with 9 seconds of a different guitar recording, calculating a normalized mean squared error for each. nMSE was defined as

$$nMSE = 10 * \log_{10} \left(\frac{(\hat{y} - y)^2}{\hat{y}^2} \right) \quad (23)$$

where y is the synthesized output and \hat{y} is the oracle output, generated by LTSpiceIV. The results are located in Table 1. A sample of the results is shown in Fig.2.

6.2. Common-Cathode Tube Amplifier

The Common-Cathode Tube Amplifier stage is an inverting voltage amplifier similar to the common emitter transistor circuit. It is a common stage in many guitar amplifiers and is valued for its overdriven sound which results in asymmetrical clipping. For the purposes of this experiment we modeled the circuit in LTSpice IV. The version we are modeling is shown in Fig.3 and includes a DC blocking capacitor C_i for the input and a bypass capacitor C_b around the

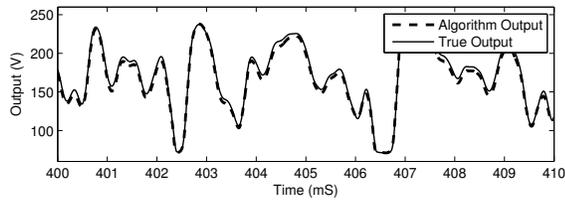


Figure 4: Comparison of our Algorithm to LTSpice Simulation of a Common-Cathode Tube Amplifier with Guitar Input.

Circuit	50 Pt.	100 Pt.	300 Pt.	500 Pt.
OCDC	-15.35 dB	-19.83 dB	-25.77 dB	-35.45 dB
CCT	-12.37 dB	-25.56 dB	-23.88 dB	-34.38 dB

Table 1: nMSE of guitar signal for two synthesized circuits

bias resistor R_b , as well as explicitly modeling the triode’s grid resistance R_g and Miller capacitance C_f . Our model will therefore include an input voltage V_i , an output voltage V_o and three state voltages, one for each of the three capacitors in the circuit. The tube itself is a Koren model of a 12AX7 tube[12]. The passive element values as referenced in Fig.3 are: $C_i = 47\mu F$, $R_i = 1M\Omega$, $R_g = 70k\Omega$, $C_f = 2.5pF$, $R_p = 100k\Omega$, $C_b = 25\mu F$, $R_b = 1.5k\Omega$.

We used the proposed algorithm to learn regression models utilizing 50, 100, 300, and 500 points for each of 3 states and the output, each of which used the Mahalanobis kernel from equation (21) whose parameters were learned via gradient descent[7]. The Add The Worst algorithm was run on a selection of 5 seconds of guitar recordings of the input, output, and each of the 3 states. For each SVM the training points were chosen and the kernel parameters were learned. We then tested the simulation by feeding it both a 9 second recording of a different guitar playing a different song. The normalized mean squared error was calculated and tabulated in Table 1. A sample of the output from the guitar test can be seen in Fig.4.

Audio examples for both the One Capacitor Diode Clipper and the Common-Cathode Tube Amplifier can be found at <http://labrosa.ee.columbia.edu/projects/nlds/>.

7. CONCLUSIONS

Although guitar distortion circuits tend to produce not easily tractable nonlinear dynamical systems, we’ve shown that the application of kernels can transform these systems into easily solvable linear dynamical systems, and solved in a regression framework. We’ve also suggested some methods for choosing the proper kernel to linearize these systems and confirmed via experimentation that these linearized dynamical systems can closely approximate the underlying nonlinear dynamical system.

Though the application of kernel regression to guitar circuits is novel, and the application of kernels to general nonlinear dynamical systems has been examined only briefly[13][14], kernel regression itself has been extensively studied and kernel selection is an active research area. Existing or new methods of kernel selection would be applicable to this work and could have the potential to improve its results. Also, much work in kernel and SVM regression has been done developing new constraints to add to the inherent SVM optimization problem, a technique which may prove useful here.

Because of the white box nature of this problem, we have broad capacity to derive more constraints based on knowledge of both the circuit elements and structure.

8. REFERENCES

- [1] J. Pakarinen and D. T. Yeh, “A review of digital techniques for modeling vacuum-tube guitar amplifiers,” *Comput. Music J.*, vol. 33, no. 2, pp. 85–100, June 2009. [Online]. Available: <http://dx.doi.org/10.1162/comj.2009.33.2.85>
- [2] D. Yeh, J. Abel, and J. Smith, “Automated physical modeling of nonlinear audio circuits for real-time audio effects:part i: Theoretical development,” *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 18, no. 4, pp. 728–737, 2010.
- [3] D. T. Yeh, J. S. Abel, A. Vladimirescu, and J. O. Smith, “Numerical methods for simulation of guitar distortion circuits,” *Comput. Music J.*, vol. 32, no. 2, pp. 23–42, June 2008. [Online]. Available: <http://dx.doi.org/10.1162/comj.2008.32.2.23>
- [4] D. T. Yeh and J. O. Smith, “Simulating guitar distortion circuits using wave digital and nonlinear state-space formulations.”
- [5] M. O. Franz and B. Schölkopf, “A unifying view of wiener and volterra theory and polynomial kernel regression,” *Neural Comput.*, vol. 18, no. 12, pp. 3097–3118, Dec. 2006. [Online]. Available: <http://dx.doi.org/10.1162/neco.2006.18.12.3097>
- [6] A. J. Smola and B. Schölkopf, “A tutorial on support vector regression,” *Statistics and Computing*, vol. 14, no. 3, pp. 199–222, Aug. 2004. [Online]. Available: <http://dx.doi.org/10.1023/B:STCO.0000035301.49549.88>
- [7] O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee, “Choosing multiple parameters for support vector machines,” *Mach. Learn.*, vol. 46, no. 1-3, pp. 131–159, Mar. 2002. [Online]. Available: <http://dx.doi.org/10.1023/A:1012450327387>
- [8] O. Chapelle and A. Rakotomamonjy, “Second order optimization of kernel parameters,” in *In NIPS Workshop on Kernel Learning*, 2008.
- [9] Y. Kamada and S. Abe, “Support vector regression using mahalanobis kernels,” in *Proceedings of the Second international conference on Artificial Neural Networks in Pattern Recognition*, ser. ANNPR’06. Berlin, Heidelberg: Springer-Verlag, 2006, pp. 144–152. [Online]. Available: http://dx.doi.org/10.1007/11829898_13
- [10] R. G. Keen, “The Technology of the Tube Screamer,” Geofex, Tech. Rep., 1998. [Online]. Available: http://www.geofex.com/Article_Folders/TStech/tsxtech.htm
- [11] T. E. Stinchcombe, “A study of the korg ms10 & ms20 filters,” 2006.
- [12] N. Koren, “Improved vacuum tube models for spice simulations,” 1996.
- [13] L. R. F. d’Alche Buc, “Dynamical modeling with kernels for nonlinear time series prediction,” *Advances in neural information processing systems*, vol. 16, p. 129, 2004.
- [14] H. Zhang, X. Wang, C. Zhang, and X. S. Cai, “Robust identification of non-linear dynamic systems using support vector machine,” *Science, Measurement and Technology, IEE Proceedings -*, vol. 153, no. 3, pp. 125–129, 2006.