A Causal Locally Competitive Algorithm for the Sparse Decomposition of Audio Signals

Adam S. Charles, Abigail A. Kressner, and Christopher J. Rozell
School of Electrical and Computer Engineering
Georgia Institute of Technology, Atlanta, Georgia 30332-0250
Email: {acharles6,abbiekre,crozell}@gatech.edu

***Abstract*—While current inference methods can decompose audio signals, they require the entire signal upfront and are therefore ill-suited for real-time applications requiring causal processing. We propose a neurally-inspired, causal, sparse inference scheme based on the Locally Competitive Algorithm (LCA) over a temporal-spectral neighborhood. We demonstrate that this causal inference scheme can achieve lower sparsity levels and better signal fidelity than current filter and threshold approaches. Additionally, for some regimes, the sparsity level approaches those of Matching Pursuit while still maintaining signal integrity.**

***Index Terms*—Locally Competitive Algorithm (LCA), causal sparse encoding, audio processing, convolutional model**

I. Introduction

Sparse coding models have recently led to state-of-the-art signal and image processing algorithms in a wide variety of situations [1]. These models treat a signal as a linear combination of elements from a (potentially overcomplete) dictionary, and the goal is to find a signal approximation using as few of these dictionary elements as possible. Recently, sparse coding has been applied to audio signals in a convolutional model, where the dictionary is composed of time shifted versions of a basic set of kernels, and the linear generative model is given by

$$\mathbf{x} = \sum_{i\in[1,N]} \sum_{m} s_i^m \phi_i(n - \tau_i^m). \qquad (1)$$

In this model the audio signal, $\mathbf{x}$, is a summation of a fixed set of basis elements, $\{\phi_i\}$, scaled by a set of weights $\{s_i^m\}$, and shifted across m possible time locations by $\{\tau_i^m\}$. Current approaches to finding the optimal set of coefficients treat the audio signal as a single large block of data, therefore requiring that the whole signal be available at once. Thus, while sparse encoding is currently a useful model, real-time applications are not possible unless a causal algorithm with low-delay is developed to infer the coefficients.

Sparse coding has also been shown to correlate well with neural encoding schemes. For example, Olshausen and Field [2] have used unsupervised learning approaches to show that the optimal dictionary elements for image encoding under a sparsity model match well with measured receptive fields in the visual cortex. Likewise for the auditory system, Smith and Lewicki [3] have used a similar approach to show that optimal dictionary elements in a convolutional model such as (1) correspond with measured auditory receptive fields (often modeled as gammachirp functions [4], [5]). Because these neural systems have shown evidence of using sparse approximation and they operate as causal real-time systems, we look to them as inspiration for developing a system that can be used in engineering applications.

This work is funded by NSF grants CCF-0905346 and CCF-0830456.

Currently, two main approaches to calculating sparse coefficients for audio signals exist: (1) global methods treating the signal as a large block and applying algorithms such as Matching Pursuit (MP) and (2) a causal Filter and Threshold (FT) method (such as that proposed by Smith and Lewicki [3], [6]) that treats the signal as a data stream and tries to induce sparsity.

A. Matching Pursuit

The problem of decomposing a vector $\mathbf{x}$ into a sparse set of coefficients is a well known problem in the signal processing community. The optimal coefficients would ideally be found by performing the optimization

$$\hat{\mathbf{s}} = \arg\min \|\mathbf{s}\|_0 \qquad s.t. \qquad \|\mathbf{x} - \tilde{\Phi}\mathbf{s}\|_2^2 < \epsilon, \qquad (2)$$

where $\tilde{\Phi}$ is defined as the full basis composed of all of the possible shifts of the basis vectors $\{\phi_i\}$, and the pseudo-norm $\|\mathbf{s}\|_0$ counts the number of non-zeros. In the reformulation used for Equation (2), the location of each element of $\mathbf{s}$ indicates which (i, m) pair that value corresponds to, thus indicating its shift as well.

The optimization problem in (2) is combinatorial in nature, meaning that to find the true minimum, every possible combination of basis elements must be tried. Matching Pursuit (MP) is a greedy algorithm designed to approximate the true solution [7]. In the MP algorithm, the residual vector $\mathbf{r}_0$ is initialized to the signal vector $\mathbf{x}$. At each iteration, the inner product between each residual basis element and the residual vector, $a_{i,n} = \langle \phi_i, \mathbf{r}_n \rangle$, is taken. Of these, the maximum inner product is found, and the set of coefficients is updated to include a new coefficient based on that maximum value. The residual is then updated by removing that portion of the signal. Each iteration

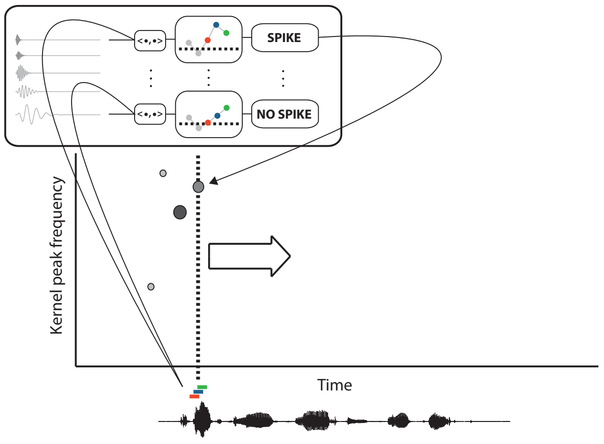

Fig. 1: Schematic illustration of the filter and threshold (FT) algorithm. Local peak detection on the inner products of three consecutive sections of the input signal yield active coefficients, which are recorded in the "spikegram".

of the algorithm can thus be summarized as

$$\mathbf{a} = \tilde{\Phi}^T \mathbf{r}_n \tag{3}$$

$$i_{max,n} = \arg\max \mathbf{a} \tag{4}$$

$$\mathbf{s}(i) = \mathbf{s}(i) + \mathbf{a}(i) \tag{5}$$

$$\mathbf{r}_{n+1} = \mathbf{r}_n - \mathbf{a}(i)\tilde{\phi}_i. \tag{6}$$

Equations (3) - (6) are repeated until the stopping criteria is met, which may be an upper bound on $\|\mathbf{r}_n\|_2^2$, a lower bound on $\max \mathbf{a}$, or the support of $\mathbf{s}$, $\|\mathbf{s}\|_0$.

B. *Filter and Threshold*

Although the MP algorithm (along with other sparse approximation algorithms applied to the block signal, including many algorithms related to MP) yield a sparse decomposition of the audio signal $\mathbf{x}$, the process requires knowledge of the entire signal and is therefore non-causal. Thus MP is ill-suited for real-time applications where future samples are unknown. Smith and Lewicki [3] proposed a simple, neurally-inspired causal method that treats the projections onto the various basis elements as an instant-by-instant projection (i.e. a filtering operation), rather than finding the optimal time shift for a dictionary element across the entire audio signal. This approach treats each filter as a neuron which responds to a specific basis element. Since neurons only respond when a critical threshold is reached, the outputs of the filters are thresholded to some value, λ, which would also enforce a level of sparsity (Figure 1).

Although the thresholding operation attempts to sparsify the output, the FT algorithm does not yield sparse results. One of the hindrances of this scheme is its tendency to have the outputs of each filter be consecutively above threshold for long stretches of time due to the correlations between successive shifts of a dictionary element (particularly those with significant low-frequency content). To further sparsify the outputs, Smith and Lewicki recommend a peak detection algorithm to choose a single value for each of these occurrences [3]. Since most peak detection algorithms are not causal, a local peak detection algorithm can be implemented by using a short buffer.

C. *Sparse and Causal Scheme*

The two algorithms discussed above (MP and FT) are each suboptimal. MP yields sparse results, yet is non-causal. The FT algorithm is causal but does not make sufficient use of the signal statistics, resulting in less sparse representations. We propose a neurally-inspired model for causal sparse coefficient recovery based on an alternate scheme of recovering sparse coefficients using a Locally Competitive Algorithm [8] (LCA). The LCA is a dynamical system that solves a family of sparse approximation problems using continuous-time nodes and analog computational primitives, making it amenable to implementation in an analog circuit that is fast and low-power compared to digital counterparts. Compared to the FT algorithm, the LCA accounts for both the sparsity measure as well as signal fidelity, which makes the LCA a better candidate for decomposing longer signals into a block-based inference problem with low delay.

II. The Causal LCA Scheme

A. *System Architecture*

Although the FT scheme provides a method for causal decompositions of audio signals, the results are not particularly sparse. Despite the added peak detection step, the fundamental problem is that successive shifts of the basis functions in time are highly correlated, producing a dictionary with very high coherence. In order to further sparsify the output coefficients, a method which takes a larger temporal neighborhood, as well as a spectral neighborhood, into account needs to be implemented. To get a sense of the required time-frequency neighborhood, the strengths of the inner products between the coefficients and their shifts were considered by observing the auto- and cross-correlation functions. Figure 2 shows two examples, $R_{\phi_3,\phi_4}(m)$ and $R_{\phi_6,\phi_4}(m)$, where ϕ_3, ϕ_4 and ϕ_6 represent gammachirp functions centered at 257.91Hz, 363.36Hz, and 647.74Hz, respectively. The correlations suggest that the most significant inner products exits within a small time-frequency window about the basis function. Basis functions either too distant in time or in frequency were much less correlated, if correlated at all.

A graphical model which shows the connections between correlated basis functions of various shifts and frequencies is shown in Figure 3. In this model, the weights along the lines which connect the nodes are the inner products between the corresponding basis functions.

The architecture shown in Figure 3 is similar to the architecture of the Locally Competitive Algorithm (LCA) [8]. The

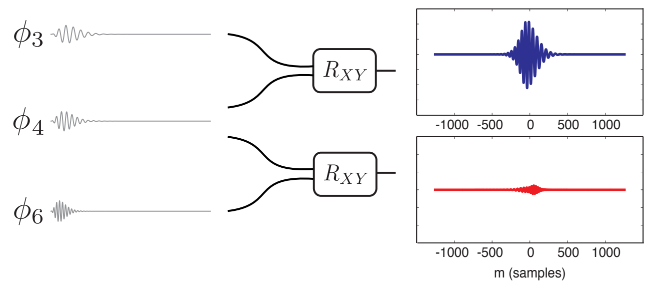

Fig. 2: Sample correlation functions between gammachirp basis elements. Basis functions ϕ_3, ϕ_4 and ϕ_6 are shown on the left. The cross-correlations, R_{ϕ_3,ϕ_4} and R_{ϕ_6,ϕ_4}, are plotted to the same scale on the right.

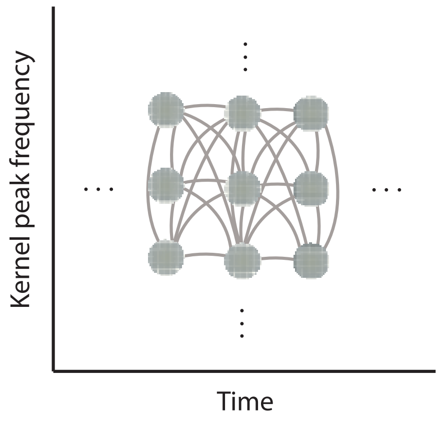

Fig. 3: Graphical model of basis correlations. Each node represents a basis function at a specific time and frequency. Connections are weighted by the inner product of the corresponding basis functions.

LCA is a neurally-inspired method which uses precisely the same weights as suggested above for the graphical model in Figure 3 to obtain sparse coefficients for linearly generative models. In the LCA algorithm, each "neuron" represents a basis function and is modeled as a leaky integrator with a thresholded output. The complete dynamics of the system at each node, i, are described in terms of the internal variable, u_i, as

$$z_i = \sum_{j \neq i} \langle \phi_i, \phi_j \rangle a_j \tag{7}$$

$$\dot{u}_i = -\frac{1}{\tau} u_i + \frac{1}{\tau} (b_i - z_i) \tag{8}$$

$$a_i = T_\lambda(u_i), \tag{9}$$

where a_i is the output of the corresponding neuron, given by the thresholding operation T_λ on u_i, b_i is a forcing function given by the inner product $\langle \phi_i, \mathbf{x} \rangle$, and z_i is the inhibition on the neuron by the other neurons, wherein each inhibition term is proportional to the inner product between the representative basis functions. Thus the connections in Figure 3 can be treated as inhibitory connections. By allowing the internal

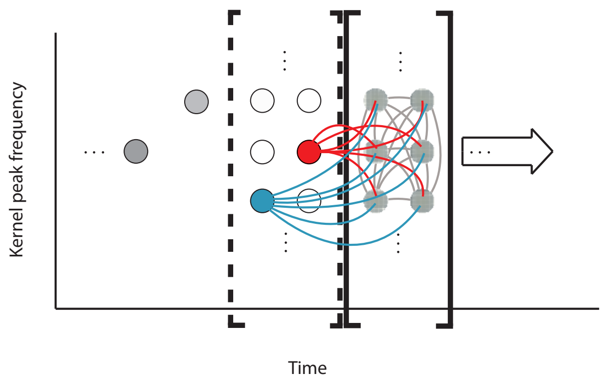

Fig. 4: Interdependencies between basis functions in a neural architecture. Coefficients for shifted bases still in the buffer remain undetermined at time n. Coefficients written recently continue to inhibit coefficients in the buffer, shown by the red and blue nodes.

variables to progress by the dynamics of Equations (7) - (9), a sparse representation can be found.

We propose extending the LCA to a causal scheme by incorporating a relevant temporal-spectral neighborhood which is identified by the correlation functions. In such a causal scheme, the last L outputs of a filtering operation with each basis element is buffered to form the significant neighborhood, as depicted in Figure 4. At each iteration the N new values coming from the filter bank (of N filters) are shifted into the buffer, and the LCA is run on the buffer. The coefficients at the output of the LCA which have been in the buffer longest (and have had the longest time to stabilize) are committed then to the decomposition as they leave the buffer. New values are shifted into the buffer again, and the process repeats.

Since the coefficients which have recently been committed still have significant correlations to coefficients which remain in the buffer, as shown in Figure 4, this causal scheme must take into account these additional correlations. Thus, the dynamics at each time step are governed by the standard LCA dynamics with a second set of inhibitions from recently committed coefficients.

B. Causal System Dynamics

The dynamics for the causal LCA (CLCA) scheme follow from the LCA dynamics. The buffer, made up of the set of b_i, is saved as the matrix $\mathbf{B} \in \mathbb{R}^{NxL}$, consisting of the past L outputs of each of the N filters in the filter bank. Thus the elements of $\mathbf{B}$ are representative of the forcing values in Equation (8): $\mathbf{B}_{i,j} = b_i(n+j) = \langle \phi_i, \vec{x}_M(n+j) \rangle$, where $\vec{x}_M$ is the truncated section of $\mathbf{x}$.

In order to fit the LCA architecture, the forcing vector $\tilde{\mathbf{b}}$ is defined as the concatenation of the transposed rows of $\mathbf{B}$. The effective basis for the LCA operation is then a concatenation of the shifted basis elements. The modified dynamics for the causal implementation of the LCA are then given by

$$\tilde{z}_i(n) = \sum_{j \neq i} \sum_{m=0}^{L-1} \langle \phi_i(n), \phi_j(n+m) \rangle a_j(n-m) + \sum_{j=1}^{N} \sum_{m=L}^{L+M-1} \langle \phi_i(n), \phi_j(n+m) \rangle a_j(n-m) \quad (10)$$

$$\dot{\tilde{u}}_i(n) = -\frac{1}{\tau}\,\tilde{u}_i(n) + \frac{1}{\tau}\left(\tilde{b}_i(n) - \tilde{z}_i(n)\right) \quad (11)$$

$$\tilde{a}_i(n) = T_\lambda(\tilde{u}_i(n)), \quad (12)$$

where the index i represents the basis function number, n represents the current time step, and M represents the range of feedback inhibitions in samples. As opposed to the inhibition given in Equation (7), the inhibition $\tilde{z}_i(n)$ in Equation (10) is composed of two sets of terms. The first set of terms is identical to Equation (7) and represents the inhibitions internal to the buffer. The second term represents the feedback inhibitions from the coefficients written in the past M time steps. Since the coefficients $a_i(n-m)$ over $m \in [L+M-1, L-1]$ are determined, this term is constant for every time step.

At steady state, the matrix $\mathbf{A}$, composed of the $\tilde{a_i}$ for each time step j, is the coefficient matrix corresponding to $\mathbf{B}$. The algorithm saves the first column of $\mathbf{A}$, which corresponds to the earliest basis elements represented in the buffer, to the final decomposition. The remaining coefficients are shifted by one, and the new column of $\mathbf{A}$ is initialized to zero for the LCA at the next time step.

III. Methods

In order to evaluate the performance for CLCA relative to Matching Pursuit and filter-and-threshold, we processed a speech sample with each of the three inference algorithms. We used the test signal "aba", with a sampling frequency of 16kHz, from the database included in [9] and bandpass filtered it from 100Hz to 6kHz. We constructed the basis set of 16 gammachirp functions logarithmically spaced from 100 Hz to 6kHz using Irino's dynamic compressive gammachirp (dcGC) filterbank software [4]. The parameters used were specified by Park [10] and inspired by Irino and Patterson [11]. Specifically, each gammachirp function was determined by the following equations.

$$g_{f_c}(t) = t^3 e^{-2\pi \cdot 1.14 \cdot B(f_c)} \cos\left(2\pi f_c \cdot t + 0.979 \cdot \ln t\right), \quad (13)$$

where $B(f_c)$ is the Equivalent Rectangular Bandwidth (ERB) of the center frequency, f_c,

$$B(f_c) = 0.1039 f_c + 24.7. \quad (14)$$

Since Equations (10) - (12) represent a continuous-time dynamical system, we implemented the system using a forward Euler's approximation. The evolution of $\tilde{u}_i(n)$ can be approximated by

$$\tilde{u}_i^k(n) = \left(1 - \frac{\Delta}{\tau}\right)\tilde{u}_i^{k-1}(n) + \frac{\Delta}{\tau}\left(\tilde{b}_i(n) - \tilde{z}_i^{k-1}(n)\right), \quad (15)$$

where k represents the iteration number and

$$\tilde{z}_i^{k-1}(n) = \tilde{z}_i(n)\big|_{\tilde{a}_i(n) = \tilde{a}_i^{k-1}(n)}, \quad (16)$$

is the inhibition calculated with the coefficients from iteration $k-1$. As the simulated system is running, we gradually lower λ from a high starting value to the final desired value according to an exponential decay. This type of continuation scheme has shown better convergence properties in many similar systems [12]. We use both the the soft-thresholding function, defined by

$$T_\lambda(u) = \begin{cases} u - \lambda & \text{if } u > \lambda \\ u + \lambda & \text{if } u < -\lambda \\ 0 & \text{otherwise} \end{cases}, \quad (17)$$

as well as the hard-thresholding function, defined by

$$T_\lambda(u) = \begin{cases} u & \text{if } u > \lambda \\ u & \text{if } u < -\lambda \\ 0 & \text{otherwise} \end{cases}. \quad (18)$$

Note that the soft-thresholding function minimizes the ℓ_1 norm of the coefficients and the hard-thresholding function minimizes the ℓ_0 norm of the coefficients, both subject to a fidelity constraint [8]. We compared our proposed approach to MP, implemented in the highly optimized Matching Pursuit Toolkit [13][1].

For the parameter selections, we chose conservative parameters in order to encourage full convergence. For the hard-thresholding we used $\Delta/\tau = 0.03$ with 4000 iterations and a decay on λ of 0.998 from an initial λ of 1. For the soft-thresholding we used a smaller $\Delta/\tau = 0.02$ with 10k iterations and the same decay on λ, but from an initial λ of 10. The buffer used in both cases was 10ms, which at a sampling rate of 16kHz is 160 samples.

Finally, we implemented the FT algorithm, as depicted by Figure 1, using local peak detection with a buffer of length three. We chose to accept a coefficient when the middle value in the buffer was greater than threshold, as well as greater than both its neighbors. FT is inherently poor at accurately representing signal strength since it ignores correlations between active coefficients. Therefore, we retroactively scaled signals during reconstruction by a number that produced the best SNR possible for the given coefficients. It is important to note that the scaling factor is not known a-priori and is not consistent across signals.

IV. Results

A. One-sparse signals

We first test the CLCA's ability to recover one-sparse signals by running the algorithm on synthetic signals containing only one dictionary element. While the CLCA is able to recover the correct one-sparse decomposition for dictionary elements with peak frequencies at or above about 273 Hz, the lowest dictionary elements yield less ideal decompositions. Specifically, the CLCA for our example cases result in a higher

[1]http://mptk.irisa.fr/

sparsity (18-50 coefficients) with signal recovery around 22 dB SNR.

The CLCA's recovery of one-sparse signals demonstrates that the CLCA retains the LCA's sparsity seeking property. The inability of the CLCA algorithm to recover the one-sparse solution for very low frequency dictionary elements, however, indicates that the high dictionary coherence is hindering the convergence of the algorithm to a significant extent.

B. Speech samples

We show the time waveform of the signal "aba" in Figure 5a and its corresponding spectrogram in Figure 5b. In Figures 5c-e, we have plotted the resulting "spikegrams" for each algorithm, where each circle corresponds to a coefficient in the sparse approximation. The size and color indicate coefficient magnitude, where large, dark-colored circles correspond to a high magnitude and small, light-colored circles correspond to a low magnitude. The scales of the circle sizes and colors are consistent across plots with the range of values falling between 0.05 and 1.7. Figure 5d, corresponding to the CLCA simulation using hard thresholding, depicts the resulting decomposition using the 10ms buffer. We specified a stopping criterion of 11.515dB signal-to-noise ratio (SNR) for MP, which matches the resulting SNR of the CLCA decomposition, and implemented the FT algorithm using local peak detection with a buffer of length three and a threshold of 0.05.

With 676 coefficients, inference using the CLCA returned an SNR of 11.515dB. Conversely, at 11.515 dB SNR, inference using Matching Pursuit returned 461 coefficients. Finally, inference using the filter and threshold algorithm returned 1835 coefficients with an SNR of 5.203dB when the reconstructed signal was scaled by 0.215 (-9.758dB without scaling). These results demonstrate the CLCA's ability to outperform the current causal method, FT, in terms of both sparsity and signal integrity. The CLCA, however, uses more coefficients than MP, demonstrating the cost of performing computations causally.

We measure the total performance in terms of the relationship between sparsity and signal integrity. We change the main parameter for each method (target SNR in MP and λ in the CLCA and FT) to obtain a range of possible fidelities for each algorithm. Figure 6 shows the parametric rate-distortion curves comparing output SNR in dB and sparsity level. While MP performes best overall, the CLCA using the hard-thresholding scheme matches the MP performance up to approximately 12dB SNR. The hard-thresholding scheme yielded unstable results for lower values of λ, limiting the feasible SNR range. Conversely, we were able to reach an SNR of 28.7dB using the soft-thresholding scheme with $\lambda = 0.0025$. Higher signal integrity may be possible with the appropriate parameter selection.

By examining the rate-distorion curves in Figure 6, it is clear that for our test signal, the CLCA can clearly outperform the FT algorithm by improving both sparsity and signal integrity. The hard-thresholding function allows the CLCA to obtain very low sparsity levels for low SNR levels, while the soft-thresholding function allows the CLCA to obtain higher SNR levels at the cost of sparisty. While not strictly real-time, the

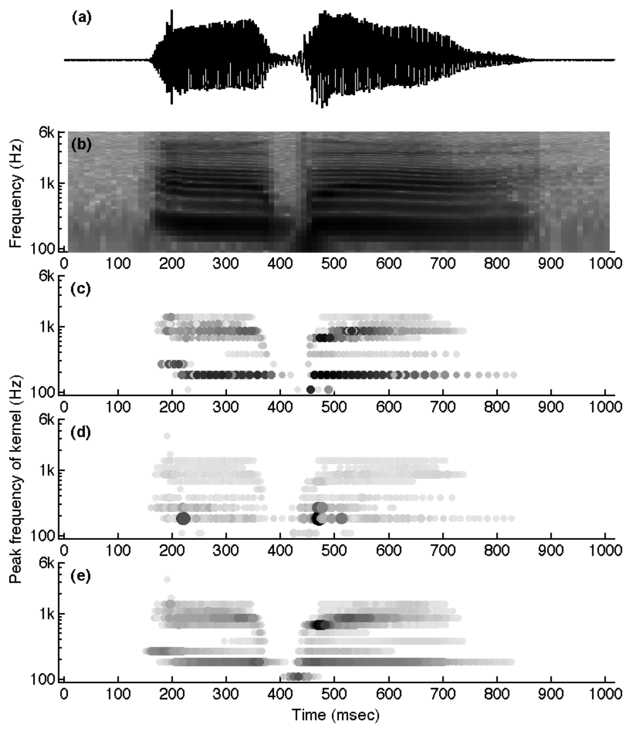

Fig. 5: (a) Sample audio file of a voice saying "aba" and (b) the corresponding power spectrum. Spikegrams for (c) MP, (d) CLCA, and (e) filter and threshold

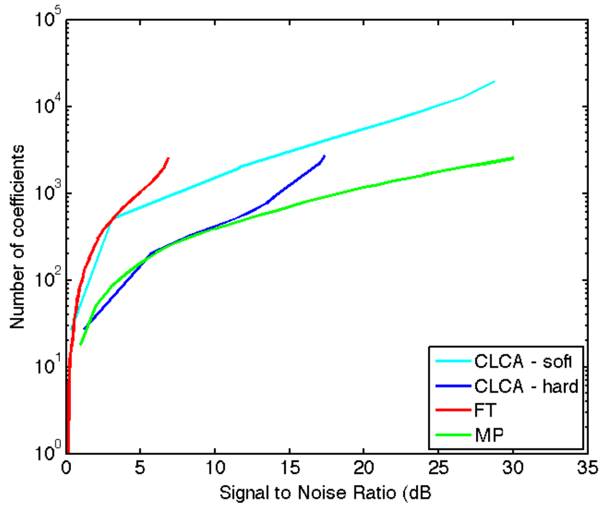

Fig. 6: Rate-distortion curves show the relationship between sparsity levels and signal SNR for various parameter choices.

algorithm is causal and runs with only a 10ms buffer.

These preliminary results suggest that with appropriate parameter selection, the CLCA could provide a near real-time solution to the sparse coding problem. While digital simulations of the CLCA are quite long, an analog implementation would be able to perform the equivalent computations much faster. With the parameters used to generate the data for Figure 6, each buffered LCA takes approximately 120 time constants. Therefore, with a system using a 100ns time-constant, the total time per buffered LCA would be 20μs. With a 16kHz sampling rate, as with the "aba" signal, the sample period is 62.5μs, which leaves sufficient time for LCA convergence. Thus, the amount of time taken for the CLCA to run would be approximately the duration of the original audio signal.

V. Conclusion

Preliminary results for the CLCA algorithm demonstrate that it is a possible causal inference scheme that can achieve relatively low sparsity levels in comparison to current methods. The results shown here are digital implementations, necessitating high simulation times. Thus analysis of only one speech signal is shown. A full analysis would include many speech signals in order to understand the scope of the CLCA algorithm.

The varying results we observe with different parameter selections indicate that we are not currently achieving proper convergence for each individual LCA. We need to more fully understand the impact of the coherence between dictionary elements in a convolutional model in order to choose LCA parameters which guarantee convergence. More informed parameter choice may improve the overall CLCA performance. Additionally, we need to quantify the detrements of the buffering scheme, irrespective of the inference algorithm used at each time step.

If future simulations are consistent with the preliminary results, the CLCA could provide a near real-time inference scheme. An analog implementation of this algorithm would be able to run real-time encoding of speech signals, making the CLCA a useful front-end for various real-time applications. Furthermore, the generality of the algorithm can extend to any time-series analysis that can be modeled by a similar convolutional model. Although there is much room for improvement, we are encouraged by these preliminary results.

References

[1] A. M. Bruckstein, D. L. Donoho, and M. Elad, "From sparse solutions of systems of equations to sparse modeling of signals and images," *SIAM Review*, vol. 51, no. 1, pp. 34–81, Feb 2007.

[2] B. A. Olshausen and D. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, vol. 381, no. 13, pp. 607–609, Jun 1996.

[3] E. C. Smith and M. S. Lewicki, "Efficient auditory coding," *Nature*, vol. 439, no. 23, pp. 978–982, Feb 2006.

[4] T. Irino and R. D. Patterson, "A dynamic compressive gammachirp auditory filterbank," in *IEEE Trans. Audio, Speech, and Language Process.*, vol. 14(6), 2006, pp. 2222–2232.

[5] L. Carney and T. Yin, "Temporal coding of resonances by low-frequency auditory nerve fibers: single-fiber responses and a population," *Journal of Neurophysiology*, vol. Volume 60, pp. 1653–77, 1988.

[6] E. C. Smith and M. S. Lewicki, "Efficient coding of time-relative structures using spikes," *Neural Computation*, vol. 17, pp. 19–45, 2005.

[7] S. Mallat, *A Wavelet Tour of Signal Processing, The Sparse Way*, 3rd ed. Elsevier, Inc, 2009.

[8] C. J. Rozell, D. H. Johnson, R. G. Baraniuk, and B. A. Olshausen, "Sparse coding via thresholding and local competition in neural circuits." *Neural computation*, vol. 20, no. 10, pp. 2526–63, Oct 2008.

[9] P. Loizou, *Speech enhancement: Theory and practice*. CRC press, Boca Raton: FL, 2007.

[10] A. Park, "Using the gammachirp filter for auditory analysis of speech," May 2003, unpublished.

[11] T. Irino and R. D. Patterson, "A time-domain, level-dependent auditory filter: The gammachirp," *J. Acoust. Soc. Am.*, vol. 101, no. 1, pp. 412 – 419, 1996.

[12] E. T. Hale, W. Yin, and Y. Zhang, "A fixed-point continuation method for l1-regularized minimization with applications to compressed sensing," Rice University Department of Computational and Applied Mathematics, Tech. Rep., Jul 2007.

[13] S. Krstulovic and R. Gribonval, "MPTK: Matching Pursuit made tractable," in *Proc. Int. Conf. Acoust. Speech Signal Process. (ICASSP'06)*, vol. 3, Toulouse, France, May 2006.