The use of least squares lattice algorithm in the parameterization and sorting of action potentials signals

José N. S. Sarinho Filho[a], Marcio Eisencraft[b], Ricardo Suyama[b], Erich T. Fonoff[c] and Maria D. Miranda[a]

[a]Departamento de Telecomunicações e Controle, Escola Politécnica da Universidade de São Paulo, São Paulo, Brazil

[b]Centro de Engenharia, Modelagem e Ciências Sociais Aplicadas, Universidade Federal do ABC, Santo André, Brazil

[c]Instituto de Psiquiatria, Faculdade de Medicina da Universidade de São Paulo, São Paulo, Brazil

ABSTRACT

The understanding of neuronal function under the action of a certain stimulus can be facilitated using techniques to distinguish the potential action from different neurons. Thus, from simultaneous recording of multiple neurons one can determine the firing patterns of each of them. Usually these techniques are implemented in three stages. From raw electrical potentials recorded using an intracranial electrode, spikes are detected, then parameterized and finally sorted, attributing every single spike observed to a particular neuron. Recently, it was proposed an on-line sorting method based on the noise level. Nevertheless, sorting is done directly based on the raw samples. In this paper we introduce an alternative way using the modified Least Squares algorithm based on the priori error with error feedback to parameterize the raw signals before classification. Preliminary simulations results show that using parameters provides performance near to results where the sorting is done directly based on the raw samples.

Keywords: Action potentials; spike sorting; parametrization; least squares lattice algorithm

1. INTRODUCTION

It is already well established that complex brain processes are reflected by the activity of neural populations.[1,2] For this reason the analysis of simultaneous action potentials from many neurons, which is achieved by extracellular recording of multiple brain sites using single wire electrodes, are widely used on the understanding of cortical activity and its mapping. On the same context in the last years has emerged a surgical treatment, called Deep Brain Stimulation (DBS), involving the implantation of a brain pacemaker, which sends electrical impulses to specific parts of the brain in order to treat movement and affective disorders such as chronic pain, Parkinson's disease, tremor and dystonia.[3] In the DBS procedure the electrical impulses are applied using implanted electrodes placed at a target region known as subthalamic nucleus and the exact positioning of stimulus electrodes plays a essential role in the successes of this treatment. The subthalamic region is previously mapped to the surgical procedure using Magnetic Resonance Imaging (MRI) which indicates the position of this region but does not indicates where the electrodes must be positioned inside it. For this purpose the local spikes firing rate is used. During the surgical implant, a empirical procedure to observe the firing rate is find with some audio device, a specific firing rate pattern, that indicates when the electrode reaches the subthalamic nucleus and a possible best region for implant the stimuli electrode. A less empirical used procedure is based on the recording

Further author information: (Send correspondence to M.E.)
N.S.: E-mail: sarinho.nivaldo@gmail.com
M.E.: E-mail: marcio.eisencraft@ufabc.edu.br
R.S.: E-mail: ricardo.suyama@ufabc.edu.br
E.T.F.: E-mail: fonoffet@usp.br
M.D.M.: E-mail: maria@lcs.poli.usp.br

Bioelectronics, Biomedical, and Bioinspired Systems V; and Nanotechnology V, edited by Ángel B. Rodríguez-Vázquez, Ricardo A. Carmona-Galán, Gustavo Liñán-Cembrano, Rainer Adelung, Carsten Ronning, Proc. of SPIE Vol. 8068, 80680Q · © 2011 SPIE · CCC code: 0277-786X/11/$18 · doi: 10.1117/12.886950

of the sequence of signals captured by the electrodes. From this sequence, consisting of the spikes with additive noise, each spike needs to be detected and attributed to a particular neuron.

As this process usually generates huge data amounts, its supervised supervision is a slow and complex task. Then, it is desirable that it can be performed in a unsupervised way. An efficient support system based on online spike sorting can be used to make it faster and more precise than in supervised way. Another desirable feature is that the whole process can be performed in real time, which allows the development of the so called "closed-loop' experiments, where is possible to immediately change experimental stimuli according to the neural response. It is worth to notice that online an real time sorting means that, each newly detected spike is sorted as soon as it is detected and a spike observed at time t is sorted only using all information observed prior to and including point of time t.

Usually these techniques for acquirement and analysis of electrophysiological data (spike sorting) are implemented in three stages.[4,5] From raw electrical potentials recorded using an intracranial electrode, spikes are detected, then parameterized and finally sorted, attributing every single spike observed to a particular neuron. Quiroga et al.[4] proposed a sorting method based on wavelet coefficients and superparamagnetic clustering. The entire process of detection, feature extraction, and clustering is performed without supervision. However, this is an off-line method and it is not able to efficiently classify spikes that recur at low frequencies. Rutishauser et al.[5] proposed a relatively simpler sorting method based on the noise level and that can be implemented on-line. The entire process is performed without supervision and relatively quickly. The algorithm iteratively updates the model and assigns spikes to clusters. It thus does not require a separate learning phase and is capable of detecting new neurons during the experiment. Nevertheless, sorting is done directly based on the raw samples. It is worth to notice that the parametrization of each detected spike can reduce the number of samples that represents it and thus facilitate the clustering process.

Considering that every neuron, when stimulated responds with a standard spike shape,[6] and taking into account the good performance of Rutishauser et al.[5] method, one can conclude that:
If two neurons have the same direction relating to the electrode, their responses to a given stimulus when reach the electrode have spikes shapes relatively close. In opposition, if two neurons have spatial orientation relating to the electrode very different, the paths from neurons to electrodes are differents and their responses to a given stimulus when reach the electrode have spikes shapes relatively differences.

Therefore, if every neuron sends a signal pattern, if the differences between the waveform from one neuron and another are far less significant that changes that It suffer to traverse the transmission medium, then different forms of spikes arriving at the electrode are mainly related to the different paths. This suggests that different forms of spikes arriving at the electrode can be classified according to the parameters that caracterize the path that each waveform traveled. Then, the problem can be reduced in a case of inverse modeling with nonstationary data. In this context, the adaptive lattice filters offers an edge by easily revealing the system.[7,8] A lattice filter of order M is a linear prediction filter with M stages. The output of each stage are the backward and forward prediction erros. The reflection coefficients define each stage of the lattice filter and can be used to represents the unknown system.[8]

In this article, we introduce an alternative way to implement the method proposed by Rutishauser et al.[5] The main difference is the clustering step, instead of the window with N samples, only M reflection coefficients are used in the sorting of spike waveform. The input of the lattice filter is the sequence of N samples of spike waveform. The backward and forward prediction erros of each lattice stage are not an end in itself but a means to represents the different directions of arrival with M reflection coefficients.

2. METHODS

From the time series, the spikes are detected and aligned. In our analysis, we consider that this step was already performed and that each spike is represented by N samples. In the proposed method, we use the backward and forward linear prediction of order M and associate the M coefficients of this prediction with the original N samples. This coefficients are then used for clustering instead of the original raw time series.

In this section, we describe the algorithm used to obtain the M reflection coefficients and revise the employed clustering technique.

2.1 Spike parameterization

Here the waveform of spike i is referred as $\{S_i\}$. This waveform is represented with N points and the amplitude at the sampling point n is denoted as $s_i(n)$ for $n = 1,\ 2,\ \ldots\ ,\ N$. In the follow to simplify the notation, the spike index i is omitted.

Each M points of a running window on the waveform of spike S is represented by the vetor

$$\mathbf{u}(n) = \left[\ s(n) \quad s(n-1) \quad \cdots \quad s(n-M+1),\ \right]^T \tag{1}$$

where the superscript T denotes transposition, $n = 1,\ 2,\ \ldots\ ,\ N$ and $s(n) = 0$ for $n \leq 0$.

From all vectors $\mathbf{u}(n)$ follow the deterministic autocorrelation matrix of the waveform $\{S\}$

$$\mathbf{\Phi}(n) = \sum_{\ell=0}^{n} \lambda^{n-\ell}\mathbf{u}(\ell)\mathbf{u}^T(\ell), \tag{2}$$

where $0 << \lambda < 1$ represents the size of the estimation horizon. It is worth to notice that $\mathbf{\Phi}(n)$ satisfies[8]

$$\sum_{\ell=0}^{n} \lambda^{n-\ell}\mathbf{u}(\ell)\mathbf{u}^T(\ell) = \mathbf{K}^T(n)\mathbf{D}(n)\mathbf{D}(n)\mathbf{K}(n). \tag{3}$$

The matrix $\mathbf{D}(n)\mathbf{K}(n)$ is referred to as the Cholesky factor of $\mathbf{\Phi}(n)$. The inverse transpose of $\mathbf{K}(n)$, i.e., the matrix $\mathbf{K}^{-T}(n)$ is upper triangular with 1's along its main diagonal; all of its elements below the main diagonal are zero. Moreover, each line represents the coefficients of the backward prediction error filter, whose order corresponds to the position of that row in the matrix. $\mathbf{D}(n)$ is a diagonal matrix, whose i^{th} element represents the square root of the backward prediction error energy of a filter of order i. Thus,

$$\mathbf{b}(n) = \mathbf{K}^{-T}(n)\mathbf{u}(n) \tag{4}$$

represents the *a posteriori* backward prediction error vector.

The elements of the vector $\mathbf{b}(n)$ can be obtained from a prediction linear using a lattice structure. Each lattice stage provides prediction errors in its output. The literature contains different versions of algorithms to obtain prediction errors from the observed sequence[8] $\{u(n)\}$. However, the modified Error Feedback - Least Squares Lattice based on a priori error, denoted as EF-LSL, presents reliable numerical properties, even in the absence of persistent excitation and when implemented in finite-precision arithmetic.[9] The prediction section of the modified EF-LSL algorithm based on a priori error[9] is summarized in the Algorithm 1. Beyond the prediction section, a joint estimation section is included in the algorithm. However as the interest here is extract the reflection coefficient from each spike waveform, only the prediction section of the algorithm is shown. The variables $(E_i^f(n),\ \eta_i,\ k_i^f(n))$ and $(E_i^b(n),\ \psi_i(n),\ k_i^b(n))$ represent respectively, energies, *a priori* prediction errors and reflection coefficients of the forward and backward predictions and $\gamma_i(n)$ are the conversion factors. The variables $(b, \bar{b}, f, \bar{f})$ were introduced to reduce the computational complexity of the algorithm. To ensure robust numerical behavior in the prediction section, it is necessary to avoid divisions by values close to zero in their computations. To this end, we add a small positive constant δ to the denominators, whose value depends on the implementation precision. The variables, which are initialized with non null values, are listed in the top of this table.

Note that for each N samples the sets $\{k_i^f(n)\}$ and $\{k_i^b(n)\}$ for $i = 1,\ 2,\ \cdots\ ,\ M$ yields. These sets represent a least squares version of the reflection coefficients of the Levinson-Durbin Recursion.[8] These coefficients can be used to revealing the unknown system. For our purpose only the set $\{k_i^f(n)\}$ (or $\{k_i^b(n)\}$) is sufficient to represent the different unknown paths traveled by each waveform. To facilitated the comparison among all the sets of M reflection coefficient we take

$$k_i(n) = k_i^f(n)/\sqrt{E_i^f(n)}. \tag{5}$$

This step is included on the last line in the Algorithm 1. Then, each set of $k_i(n)$ for $i = 1,\ 2,\ \ldots\ ,\ M$ represents N samples of the spike waveform.

The computational cost to represents each set of the N samples into a set of M samples is $12 \times M \times N$ multiplications, $3 \times M \times N$ divisions, and $9 \times M \times N$ additions. For exemple, when $M = 3$ the computational cost is $36 \times N$ multiplications, $9 \times N$ divisions, and $27 \times N$ additions.

Initialization:
$\lambda = 1 - 1/N$; size of the estimation horizon
$\{E_i^f(0) = E_i^b(-1) = \lambda\}$, $i = 0, \ldots, M-1$

for $n = 1, 2, 3, ..., N$ do:
  $\eta_0 = \psi_0(n) = s(n)$
  $\gamma_0 = 1$
  for $i = 0 : M-1$,
    $b = \psi_i(n-1)\gamma_i$
    $f = \eta_i \gamma_i$
    $E_i^b(n-1) = \lambda E_i^b(n-2) + \psi_i(n-1)\, b$
    $E_i^f(n) = \lambda E_i^f(n-1) + \eta_i\, f$
    $\bar{b} = b/(\delta + E_i^b(n-1))$
    $\bar{f} = f/(\delta + E_i^f(n))$
    $\gamma_{i+1} = \gamma_i - \bar{b}\, b$
  Lattice:
    $\psi_{i+1}(n) = \psi_i(n-1) - k_i^b(n-1)\eta_i$
    $\eta_{i+1} = \eta_i - k_i^f(n-1)\psi_i(n-1)$
    $k_{i+1}^f(n) = k_i^f(n-1) + \eta_{i+1}\bar{b}$
    $k_{i+1}^b(n) = k_i^b(n-1) + \bar{f}\psi_{i+1}(n)$
    $k_{i+1}(n) = k_{i+1}^f(n)/\sqrt{E_i^f(n)}$; Parameter of interest
end

Algorithm 1: Summary the prediction section of the LSL algorithm

2.2 Online spike sorting method

Following the same idea presented by Rutishauser et al.,[5] each newly detected spike is classified as soon as it is detected. The decision whether to assign the detected spike to a particular cluster or to create a new group is based on a distance function, which quantifies how similar the new pattern is to a cluster. Nevertheless, instead of comparing patterns based on the raw waveform, as in,[5] we compare the newly detected spike and the representatives of each cluster in terms of their parameters, obtained by the method discussed in Section 2.1. Hence, each newly detected spike is represented by a parameter vector $\boldsymbol{\alpha}$.

The first step in the sorting procedure is to determine if the detected spike is similar to one of the already-established clusters, which, in our case, are represented by vectors $\mathbf{c}_j$, defined as the mean value of the parameter vectors associated with spikes assigned to cluster j. If the distance function $\text{dist}(\boldsymbol{\alpha}, \mathbf{c}_j)$ is smaller than a certain threshold T_S, then the spike should be assigned to the closest cluster. If this is not the case, the new spike will form a new cluster. In our case, the distance function is given by the Euclidean distance.

Afterwards, since the addition of a spike to a cluster may alter the value of the representative vector $\mathbf{c}_j$, it may occur that the distance between two clusters, given by $\text{dist}(\mathbf{c}_i, \mathbf{c}_j)$, be significantly reduced. This may indicate that these clusters are associated with spikes that are similar, or even indistinguishable, from each other. Hence, if $\text{dist}(\mathbf{c}_i, \mathbf{c}_j)$ falls below a threshold T_M, the i-th and j-th clusters should be merged into a single cluster. The complete sorting method is summarized in Algorithm 2.

For each newly detected spike, do:

1. Extract parameters $\boldsymbol{\alpha}$ for the detected spike;
2. Evaluate $D = \min \text{dist}(\boldsymbol{\alpha}, \mathbf{c}_i)$, where $\text{dist}(\boldsymbol{\alpha}, \mathbf{c}_i)$ is a distance function and $\mathbf{c}_i$ denotes the i-th cluster center;
3. If $D > T_S$, where T_S is a predefined threshold, then create a new cluster, and return to Step 2; Otherwise, assign $\boldsymbol{\alpha}$ to the closest cluster and update the center $\mathbf{c}_i$);
4. Verify if the distances between every pair of clusters is smaller than T_M. In other words, if $\text{dist}(\mathbf{c}_i, \mathbf{c}_j) \leq T_M$ then merge clusters i and j; Otherwise, return to Step 1.

Algorithm 2: Spike sorting algorithm.

From the description above it is clear that T_S and T_M play a central role in the algorithm. In order to obtain reasonable values for T_S and T_M one should observe that all clusters are directly affected by the measurement noise, defining how disperse will be the patterns assigned to the same cluster. Therefore, after preliminary simulations, we chose $T_S = T_M = 3\sigma_n^2$, where σ_n^2 denotes the noise variance.

In order to estimate the computational cost of the clustering procedure, one should notice that the algorithm is entirely based on the Euclidean distance, and the clustering cost can be roughly estimated considering the two main tasks: measuring the distances between the new pattern and all the previously established clusters and the distances between the recently modified cluster and all other clusters. Considering that, in steady state, K clusters have been established, it is possible to obtain the costs estimates shown in Tab. 1.

Table 1. Computational Cost Estimates for the Parametrization and Clustering Tasks.

	Parametrization	Clustering	Total Cost ($N = 256, M = 3, K = 10$)
Addition	$9MN$	$(2K-1)(2M-1)$	7007
Multiplication	$12MN$	$(K-1)(M-1)$	9234
Division	$3MN$	0	2304

From Tab. 1 it is clear that the clustering cost is proportional to the number of parameters M considered for the classification task. Hence, the cost associated with clustering based on parameterized data ($M = 3$) should be significantly less than that based on raw data ($M = N = 256$).

3. RESULTS AND DISCUSSION

To study the influence of the parametrization on the sorting method described in the previous sections, we have performed simulations using the data utilized by Rutishauser et al.[5] The spikes consist of $N = 256$ samples, obtained with a sampling rate of 100kHz.

As we are interested in comparing sorting using or not parametrization, we consider here that the detection step was successfully performed in all simulations and the detected spikes are perfectly aligned. We also consider that the detected spikes are corrupted by Additive White Gaussian Noise (AWGN), a plausible assumption as argued by Rutishauser et al.[5]

For sake of comparison, besides the parameterization described in Section 2.1 and the raw time series, we use parameters obtained from Singular Value Decomposition (SVD), one of the most commonly used parameterization in off-line methods.[4]

During simulations, we noted that spikes that are very similar but with opposite signals have very close parameters, what cause sorting errors. Thus, we included the maximum value of each spike to the paramater vector to aid the clustering step.

We used two sets of action potentials:

1. Case (i): raw data is composed of action potentials coming from three distinguishable neurons (Fig. 4 of Rutishauser et al.[5]).
2. Case (ii): raw data is composed of action potentials coming from five distinguishable neurons (Fig. 7 of Rutishauser et al.[5]).

Figure 1 shows some detected spikes with a Signal-to-Noise Ratio (SNR) of 20dB for Cases (i) and (ii).

Figure 2 shows the obtained parameters for 5000 spikes of Cases (i) and (ii). In each case, the distribution considering the Lattice parameters of Eq. (5) and SVD parameters[4] are shown. The points were colored using the sorting method of Rutishauser et al.[5] using the respective parameters as input.

To compare the performance of sorting using parametrization or the row data, we apply the classification technique of Rutishauser et al.[5] for 100000 spikes for different SNR values. Figure 3 shows the error rate and the number of obtained clusters for the two studied cases as a function of the SNR. The error rate was evaluated supposing that the sorting correctly decides the number of clusters.

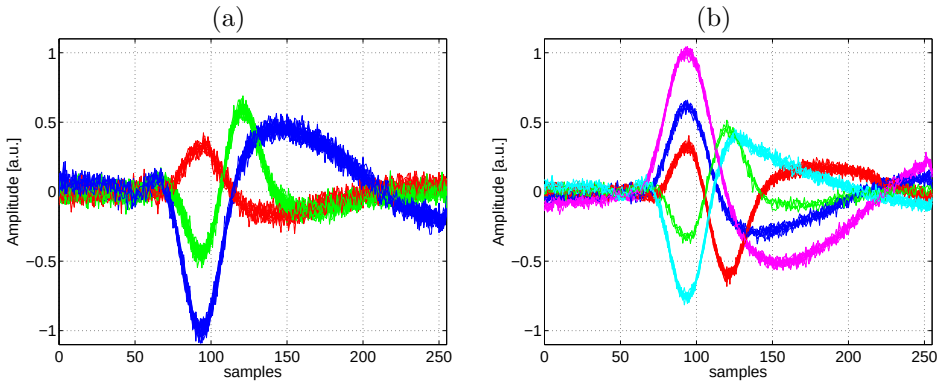

Figure 1. (a) Case (i) spike waveforms; (b) Case (ii) spike waveforms. In both cases SNR = 20dB

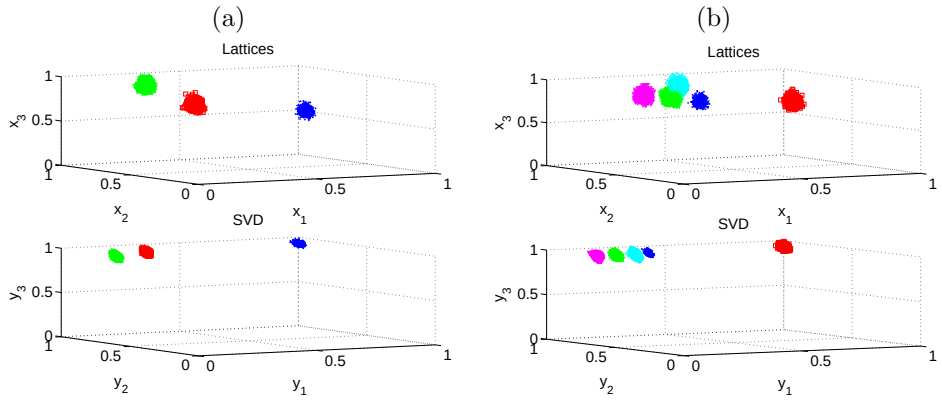

Figure 2. Parameters used as input to spike sorting technique : (a) Case (i); (b) Case (ii). In both cases SNR = 20dB.

In both cases, the simulation results show that the error rates when using the raw data are lower that those in the parameterized cases. However, for high values of the SNR, when the number of clusters is correct, both parametrization and the clustering using raw data attain similar error performance. This way, it seems that in this problem, to parameterize the data has not improved the performance of the sorting process, but it has not hampered it either, besides permitting a reduction of the data used in the clustering.

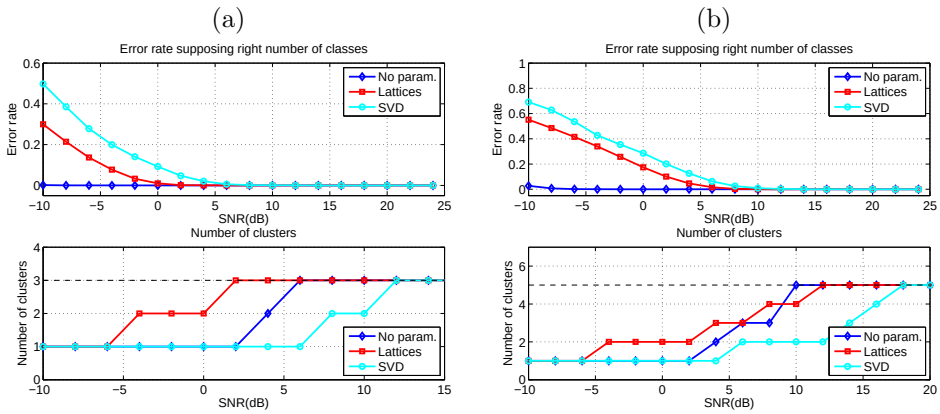

Figure 3. Error rate and number of obtained cluster as a function of the SNR (a) Case (i); (b) Case (ii). The correct number of clusters in each case is represented by dashed lines.

4. CONCLUSIONS, RELEVANCE AND FUTURE IMPROVEMENTS

In this work, it is proposed to parameterize raw action potentials signals using the modified Least Squares algorithm based on the linear prediction before classification. The proposed parametrization scheme greatly reduce the number of parameters used for spike classification. The results indicate that this approach is better than a parametrization based on SVD. However, for low values of SNR, the proposed methodology is still outperformed by the classification considering the raw data. It is important to highlight that the comparison provided in the present paper considers that all other tasks involved in the spike sorting problem are successfully accomplished. Despite the fact that the prediction-based parametrization did not exhibit good results for low SNRs, it is still possible that it can be helpfull in these other tasks, e.g., spike detection in low SNRs and spike alignment, two aspects to be included as perspectives for future works on the subject.

REFERENCES

1. E. N. Brown, R. E. Kass, and P. P. Mitra, "Multiple neural spike train data analysis: state-of-the-art and future challenges," *Nature Neuroscience* (7), pp. 456–461, 2004.
2. G. Buzsaki, "Large-scale recording of neuronal ensembles," *Nature Neuroscience* (7), pp. 446–451, 2004.
3. M. L. Kringelbach, N. Jenkinson, S. L. F. Owen, and T. Aziz, "Translational principles of deep brain stimulation," *Nat Rev Neurosci* (8), pp. 623–635, 2007.
4. R. Q. Quiroga, Z. Nadasdy, and Y. Ben-Shaul, "Unsupervised spike detection and sorting with wavelets and superparamagnetic clustering," *Neural Comput.* **16**(8), pp. 1661–1687, 2004.
5. U. Rutishauser, E. Schuman, and A. Mamelak, "Online detection and sorting of extracellularly recorded action potentials in human medial temporal lobe recordings, in vivo," *Journal of Neuroscience Methods* **154**, pp. 204–224, June 2006.
6. P. Dayan and L. Abbott, *Theoretical Neuroscience Computational and Mathematical Modeling of Neural Systems*, The MIT Press, Cambridge, MA, first ed., 2005.
7. A. Mahalanabis, S. Prasad, and K. Mohandas, "Deconvolution of nonstationary seismic data using adaptive lattice filters," *Acoustics, Speech and Signal Processing, IEEE Transactions on* **31**, pp. 591 – 598, June 1983.
8. S. Haykin, *Adaptive Filter Theory*, Prentice-Hall, New Jersey, third ed., 1996.
9. M. Miranda, M. Gerken, and M. Da Silva, "Efficient implementation of error-feedback LSL algorithm," *Electronics Letters* **35**, pp. 1308 –1309, Aug. 1999.