DECIMATED LEAST MEAN SQUARES FOR FREQUENCY SPARSE CHANNEL ESTIMATION

Omid Taheri   *Sergiy A. Vorobyov*

Dept. Electrical and Computer Engineering, University of Alberta
Edmonton, AB, T6G 2V4, Canada
Emails: otaheri, svorobyo@ualberta.ca

ABSTRACT

The standard least mean squares (LMS) parameter estimation method does not assume any special structure for the parameters being estimated. However, when additional knowledge about the system is available, the performance of LMS can be improved by appropriate modification of the algorithm. We develop such modifications for the case of estimating frequency sparse channels. Such modifications provide either better performance or less complexity when compared to the standard LMS algorithm. Decimated LMS and zero attracting decimated LMS are the two methods proposed in this paper. Simulation results are also provided to compare the performance of the proposed algorithms to the standard LMS and other sparsity aware modifications of LMS.

Index Terms— Least mean squares, sparsity, compressed sensing.

1. INTRODUCTION

The least mean squares (LMS) algorithm is a well known and widely used method for parameter estimation [1], [2]. Among its many applications are system identification and channel estimation for communication systems. The latter is the application considered in this paper. The standard LMS algorithm does not use any specific structural information about the parameters being estimated. However, if such structural information on the parameters is known, it may be possible to improve the performance of the LMS algorithm by modifying it according to this additional knowledge. There has been some work done to improve the LMS algorithm by exploiting the sparsity of systems and signals [3]– [6]. The way the problem is approached in these works is to add a penalty term to the traditional LMS cost function. This penalty term is chosen so to force the resulting signal to be sparse.

Sparsity aware modifications of other estimation algorithms such as recursive least squares (RLS) and Kalman filter have also been proposed in recent years [7], [8]. These algorithms originate from the theory of compressed sensing (CS). CS is the theory of sparse signal recovery from fewer samples than signal's dimension [9], [10]. Compressed samples are usually collected using random measurement matrices which are universal, meaning that regardless of the sparsity basis the signal can be recovered from its samples collected through such random measurement matrix. For some sparsity bases such as the frequency domain or the discrete cosine transform (DCT) domain, one can just collect random time samples of the signal and still be able to recover the signal with CS recovery algorithms [11].

In this paper, we propose LMS-based channel estimation methods for frequency or DCT sparse channels that can be recovered from a smaller number of random time samples. The main idea of this work is to update the estimate of only a subset of the channel impulse response (CIR) taps. The particular subsets chosen here are the even or odd indexed taps of the CIR. This leads to simplification of the channel estimation algorithm compared to the standard LMS.

2. BACKGROUND AND SYSTEM MODEL

The CS theory suggests that one can recover a sparse signal from a smaller number of samples than what is required by the Nyquist rate [9], [10]. In the case of discrete time signals of finite dimension, signals can be recovered from a number of samples less than the signals' dimension. Let $\boldsymbol{f}$ be a sparse vector of dimension N represented in the orthonormal sparsity basis $\boldsymbol{\Psi}$ as

$$\boldsymbol{f}_{\Psi} = \boldsymbol{\Psi}\boldsymbol{f}. \tag{1}$$

For $\boldsymbol{f}$ to be sparse in $\boldsymbol{\Psi}$, its representation $\boldsymbol{f}_{\Psi}$ has to have a few nonzero coefficients. An S-sparse signal in $\boldsymbol{\Psi}$ domain is a signal with at most S nonzero coefficients in that domain. CS suggests that a universal measurement matrix, which works for all S-sparse signals regardless of the sparsity basis $\boldsymbol{\Psi}$ can be designed [10]. Particularly, compressed samples can be taken from $\boldsymbol{f}$ using a $K \times N$ random measurement matrix $\boldsymbol{\Phi}$. The vector of compressed samples $\boldsymbol{s}$ can be expressed as $\boldsymbol{s} = \boldsymbol{\Phi}\boldsymbol{f} = \boldsymbol{\Phi}'\boldsymbol{f}_{\Psi}$ where $\boldsymbol{\Phi}' = \boldsymbol{\Phi}\boldsymbol{\Psi}^{-1}$. If a sufficient number of samples is collected, then $\boldsymbol{f}_{\Psi}$ can be obtained by solving the basis pursuit optimization problem [10], which is a convex optimization problem and can be solved by linear programing.

Supported in part by the Natural Science and Engineering Research Council (NSERC) of Canada.

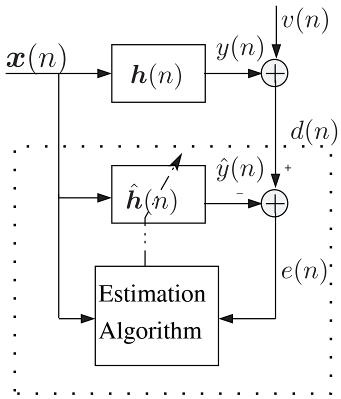

Fig. 1. Block diagram of a communication system including channel estimation block.

We consider the common problem of channel estimation in a communication system. The block diagram of the system with the channel estimation block included is shown in Fig. 1. The dotted box in Fig. 1 is the channel estimation block which is implemented and executed at the receiver. The data sequence transmitted over the channel is denoted as $\boldsymbol{x}(n)$. The vector $\boldsymbol{h}(n) = (h(n), h(n-1), \ldots, h(n-N+1))^T$ is the CIR which has a finite length N, $(\cdot)^T$ denotes the transposition operator, and $v(n)$ is an unknown channel noise at the receiver. The estimate of the channel $\hat{\boldsymbol{h}}(n)$ is updated according to a particular estimation algorithm and $e(n) = d(n) - \hat{\boldsymbol{h}}^T(n)\boldsymbol{x}(n)$ denotes the instantaneous error with $\hat{y}(n) = \hat{\boldsymbol{h}}^T(n)\boldsymbol{x}(n)$ being the estimate of the system output $y(n)$.

In standard LMS, gradient descent is used to minimize a cost function of the form $L(n) = (1/2)e^2(n)$ [2]. Therefore, the update equation of LMS can be derived as

$$\hat{\boldsymbol{h}}(n+1) = \hat{\boldsymbol{h}}(n) - \mu\frac{\partial L(n)}{\partial \hat{\boldsymbol{h}}(n)} = \hat{\boldsymbol{h}}(n) + \mu e(n)\boldsymbol{x}(n) \quad (2)$$

where μ is the algorithm's step size chosen from the range $0 < \mu < \lambda_{\max}^{-1}$ and $\lambda_{\max}$ is the maximum eigenvalue of the covariance matrix of $\boldsymbol{x}(n)$.

For the case when CIR is time sparse, i.e., when most of the coefficients in the vector $\boldsymbol{h}(n)$ are zeros, different modifications of the LMS algorithm are available [3], [6]. One of the methods presented, the zero attracting LMS (ZA-LMS), penalizes the non-sparse solutions by adding the l_1-norm of $\hat{\boldsymbol{h}}(n)$ to the standard LMS cost function. This penalty term forces the tap values of $\hat{\boldsymbol{h}}(n)$ to approach zero, and hence the name zero attracting LMS. The corresponding cost function is $L_{ZA}(n) = (1/2)e^2(n) + \gamma_{ZA}\|\hat{\boldsymbol{h}}(n)\|_{l_1}$ where $\|\cdot\|_{l_1}$ denotes the l_1-norm of a vector and γ_{ZA} is the weight associated with the penalty term. The update equation for this method can be derived as

$$\hat{\boldsymbol{h}}(n+1) = \hat{\boldsymbol{h}}(n) + \mu e(n)\boldsymbol{x}(n) - \rho_{ZA}\text{sgn}(\hat{\boldsymbol{h}}(n)) \quad (3)$$

where $\rho_{ZA} = \mu\gamma_{ZA}$ and $\text{sgn}(\cdot)$ is the sign function which operates on every component of the vector separately and $\text{sgn}(x)$ is zero for $x = 0$, 1 for $x > 0$, and -1 for $x < 0$.

Note that although only time domain sparsity is considered in [3], ZA-LMS can be easily extended to arbitrary sparsity bases. Let $\boldsymbol{\Psi}$ be the $N \times N$ orthonormal matrix denoting a specific sparsity basis. The CIR $\boldsymbol{h}(n)$ is sparse in the sparsity domain $\boldsymbol{\Psi}$ if its representation in $\boldsymbol{\Psi}$, that is, the vector $\boldsymbol{h}_\Psi(n) = \boldsymbol{\Psi}\boldsymbol{h}(n)$, has only a few nonzero components. The ZA-LMS cost function can be rewritten as $L_{ZA}(n) = (1/2)e^2(n) + \gamma_{ZA}\|\boldsymbol{\Psi}\hat{\boldsymbol{h}}(n)\|_{l_1}$, and the update equation becomes

$$\hat{\boldsymbol{h}}(n+1) = \hat{\boldsymbol{h}}(n) + \mu e(n)\boldsymbol{x}(n) - \rho_{ZA}\text{sgn}(\boldsymbol{\Psi}\hat{\boldsymbol{h}}(n))\boldsymbol{\Psi} \quad (4)$$

where $\text{sgn}(\boldsymbol{\Psi}\hat{\boldsymbol{h}}(n))$ is a row vector and $\text{sgn}(\boldsymbol{\Psi}\hat{\boldsymbol{h}}(n))\boldsymbol{\Psi}$ is a row vector as well.

In [4], variations of the ZA-LMS and RZA-LMS algorithms of [3], in which the filter coefficients are updated in a transform domain leading to faster convergence with non-white system inputs, have been proposed. A simplified version of RZA-LMS obtained through the use of piece-wise approximation has been presented in [5].

3. PROPOSED ALGORITHMS

The signals of interest in this work are those that are sparse in a domain such that the signals can be recovered from random time samples. We aim at exploiting this feature in order to design less complex or more accurate variations of the LMS algorithm. The basic idea is to estimate at even time-steps only the even taps of the CIR and do so for the odd taps at odd time-steps. To this end, the training sequence is chosen in a way that nothing is being sent into the channel at every other time-step. For example, a sample training sequence with binary phase shift keying (BPSK) symbols is $\boldsymbol{x} = 1, 0, -1, 0, -1, 0, 1, 0, 1, 0, \cdots$. In this way, at odd time-steps only the odd taps of the channel contribute in the received symbol $d(n)$ and the even taps only affect $d(n)$ at even time-steps.

We develop two variations of the LMS algorithm based on the aforementioned idea. Note however that other channel estimation algorithms can be modified based on this idea and the LMS algorithm is selected only as a popular example. According to the first algorithm, only even or odd taps have to be updated, and at the end of the training phase, an l_1-norm minimization problem is solved to estimate all channel taps. The other algorithm is an adaptation of the ZA-LMS which alternatively updates the even and odd channel taps at each time-step.

3.1. Decimated LMS

Let $\boldsymbol{r}_o(n)$ be the set of odd indexed coefficients of the vector $\boldsymbol{r}(n)$ and $\boldsymbol{r}_e(n)$ denotes the even indexed entries of $\boldsymbol{r}(n)$. The training data sequence $\boldsymbol{x}$ is designed so that a BPSK symbol is being sent into the channel at odd time-steps and nothing is sent at even time-steps. Therefore, at odd time-steps when $n = 2i + 1$, we have $y(n) = \boldsymbol{h}^T(n)\boldsymbol{x}(n) = \boldsymbol{h}_o^T(n)\boldsymbol{x}_o(n)$, since $\boldsymbol{x}_e(n)$ will be an all zero vector. Also, at even time-steps when $n = 2i$, $y(n) = \boldsymbol{h}_e^T(n)\boldsymbol{x}_e(n)$. In the decimated LMS algorithm, either the even or odd taps of the CIR are being estimated. Let us assume that an estimate of the odd taps is to be obtained. Then at every odd time-step the following LMS type update rule is used

$$\hat{\boldsymbol{h}}_o(n+2) = \hat{\boldsymbol{h}}_o(n) + \mu e(n)\boldsymbol{x}_o(n) \tag{5}$$

where $e(n) = d(n) - \hat{\boldsymbol{h}}_o^T(n)\boldsymbol{x}_o(n)$. At the end of the training process, the LMS estimate of the CIR's odd taps $\hat{\boldsymbol{h}}_o(n_f)$ is available where n_f is the final training time-step. The estimate of CIR $\hat{\boldsymbol{h}}(n_f)$ is obtained as $\hat{\boldsymbol{h}}(n_f) = \boldsymbol{\Psi}^{-1}\hat{\boldsymbol{h}}_\Psi(n_f)$ where $\hat{\boldsymbol{h}}_\Psi(n_f)$ is the solution to the following minimization problem

$$\min \|\tilde{\boldsymbol{h}}\|_{l_1} \quad \text{subject to} \quad \left\| \left(\boldsymbol{\Psi}^{-1}\tilde{\boldsymbol{h}}\right)_o - \hat{\boldsymbol{h}}_o(n_f) \right\|_{l_2} \leq \beta \tag{6}$$

where $\left(\boldsymbol{\Psi}^{-1}\tilde{\boldsymbol{h}}\right)_o$ denotes the odd indexed entries of $\boldsymbol{\Psi}^{-1}\tilde{\boldsymbol{h}}$ and β is some positive number. The above equation is the l_1-norm minimization problem for the case of noisy compressed samples [10].

The decimated LMS deals with vectors of a smaller size than the standard LMS. In the case when the total number of CIR taps, i.e., the cardinality of $\hat{\boldsymbol{h}}(n)$ is even, the size of the vectors in the decimated LMS is exactly half the standard LMS. In addition, the decimated LMS is only executed at odd or even time-steps depending on whether it estimates $\hat{\boldsymbol{h}}_o(n)$ or $\hat{\boldsymbol{h}}_e(n)$. Therefore, the decimated LMS is run only half the times that the standard LMS is executed. Considering the size of the vectors involved as well as the number of iterations that each algorithm needs for estimating CIR, we can conclude that the complexity of the decimated LMS for estimating $\hat{\boldsymbol{h}}_o(n)$ or $\hat{\boldsymbol{h}}_e(n)$ is about a fourth of the standard LMS. However, decimated LMS has to solve an l_1-norm minimization problem at the end of the training process to obtain the estimate of the complete CIR $\hat{\boldsymbol{h}}(n)$.

3.2. Zero Attracting Decimated LMS

Motivated by the fact that ZA-LMS has better performance than the standard LMS, we also present the zero attracting decimated LMS (ZAD-LMS) algorithm. In this method both even and odd taps of the CIR are being estimated. In order to force sparsity of the CIR, a term similar to the one in the update equation for the ZA-LMS in (4) is present in ZAD-LMS. Let us first consider odd taps of the CIR. For ZAD-LMS, the terms $\hat{\boldsymbol{h}}(n+1)$ and $\hat{\boldsymbol{h}}(n)$ in (4) are replaced with $\hat{\boldsymbol{h}}_o(n+2)$ and $\hat{\boldsymbol{h}}_o(n)$, while $\mu e(n)\boldsymbol{x}_o(n)$ replaces $\mu e(n)\boldsymbol{x}(n)$. Now we only need to replace $\rho_{ZA}\text{sgn}(\boldsymbol{\Psi}\hat{\boldsymbol{h}}(n))\boldsymbol{\Psi}$ with a similar term that results in a vector of the same size as $\hat{\boldsymbol{h}}_o(n)$. We choose

---

Algorithm 1 The ZAD-LMS Algorithm

Input: Data sequence $\boldsymbol{x}(n)$ and observations $d(n)$, $n = 1, \cdots, n_f$.
Output: The estimated channel $\hat{\boldsymbol{h}}(n_f)$.
1. Initialize by equating $\tilde{\boldsymbol{h}}(1)$ and $\tilde{\boldsymbol{h}}(2)$ to all zero vector.
2. Perform n_f iterations of ZAD-LMS:
for $n = 1$ to n_f **do**
  if n is odd, **then**
    Update $\tilde{\boldsymbol{h}}(n)$ with $\hat{\boldsymbol{h}}_o(n-2)$ and $\hat{\boldsymbol{h}}_e(n-1)$.
    Run the ZAD-LMS update equation (7a).
  else
    Update $\tilde{\boldsymbol{h}}(n)$ with $\hat{\boldsymbol{h}}_o(n-1)$ and $\hat{\boldsymbol{h}}_e(n-2)$.
    Run the ZAD-LMS update equation (7b).
  end if
end for
3. Form $\hat{\boldsymbol{h}}(n_f)$ using $\hat{\boldsymbol{h}}_o(n_f)$ and $\hat{\boldsymbol{h}}_e(n_f-1)$ or $\hat{\boldsymbol{h}}_e(n_f)$ and $\hat{\boldsymbol{h}}_o(n_f-1)$.

$\rho_{ZAD}\text{sgn}(\boldsymbol{\Psi}\tilde{\boldsymbol{h}}(n))\boldsymbol{\Psi}_o$ where $\boldsymbol{\Psi}_o$ is a sub-matrix of $\boldsymbol{\Psi}$ with only odd indexed columns retained. Note that $\tilde{\boldsymbol{h}}(n)$ is also used instead of the current CIR $\hat{\boldsymbol{h}}(n)$, where $\tilde{\boldsymbol{h}}(n)$ is set to an all zero vector when the algorithm starts at $n = 1$ time-step and also for $n = 2$. For every $n = 2i + 1 > 2$, $\tilde{\boldsymbol{h}}(n)$ has its odd indexed components set equal to $\hat{\boldsymbol{h}}_o(n-2)$ and its even indexed components are set equal to $\hat{\boldsymbol{h}}_e(n-1)$. When $n = 2i > 2$, $\hat{\boldsymbol{h}}_o(n-1)$ and $\hat{\boldsymbol{h}}_e(n-2)$ are used to form the vector $\tilde{\boldsymbol{h}}(n)$. Therefore, the set of update equations of the ZAD-LMS can be written as

$$\hat{\boldsymbol{h}}_o(n+2) = \hat{\boldsymbol{h}}_o(n) + \mu e(n)\boldsymbol{x}_o(n) - \rho_{ZAD}\text{sgn}(\boldsymbol{\Psi}\tilde{\boldsymbol{h}}(n))\boldsymbol{\Psi}_o, \quad n = 2i+1. \tag{7a}$$

$$\hat{\boldsymbol{h}}_e(n+2) = \hat{\boldsymbol{h}}_e(n) + \mu e(n)\boldsymbol{x}_e(n) - \rho_{ZAD}\text{sgn}(\boldsymbol{\Psi}\tilde{\boldsymbol{h}}(n))\boldsymbol{\Psi}_e, \quad n = 2i. \tag{7b}$$

At the end of training, i.e., at the n_f-th time-step, the two vectors $\hat{\boldsymbol{h}}_o(n_f)$ and $\hat{\boldsymbol{h}}_e(n_f-1)$ or $\hat{\boldsymbol{h}}_e(n_f)$ and $\hat{\boldsymbol{h}}_o(n_f-1)$, depending on n_f being odd or even, produce the estimate of CIR $\hat{\boldsymbol{h}}(n_f)$. The ZAD-LMS method is summarized in Algorithm 1.

4. SIMULATION RESULTS

In our first simulation example, the problem of estimating a CIR of length $N = 16$ is considered. The sparsity domain is the DCT domain. The sparsity level of the CIR is 1, i.e., only one of the coefficients of its representation in $\boldsymbol{\Psi}$ is nonzero. This nonzero coefficient takes the value of either 1 or -1 with the same probability.

Alongside the proposed algorithms, i.e., the decimated LMS and ZAD-LMS, the performance of the standard LMS and ZA-LMS methods are also measured as a point of reference. Parameter choices for these algorithms are $\rho_{ZA} =$

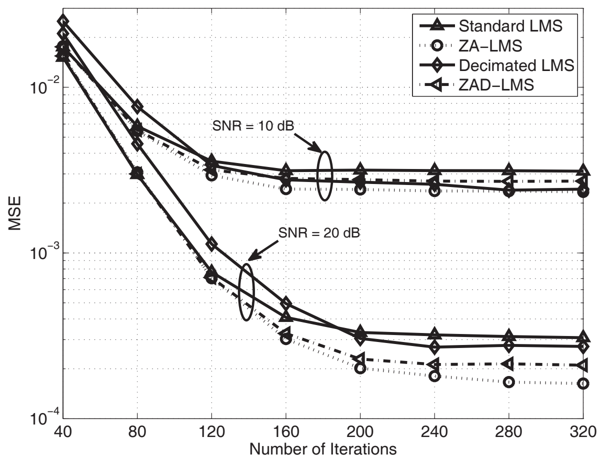

Fig. 2. Simulation example 1: Performance comparsion of different estimation algorithms.

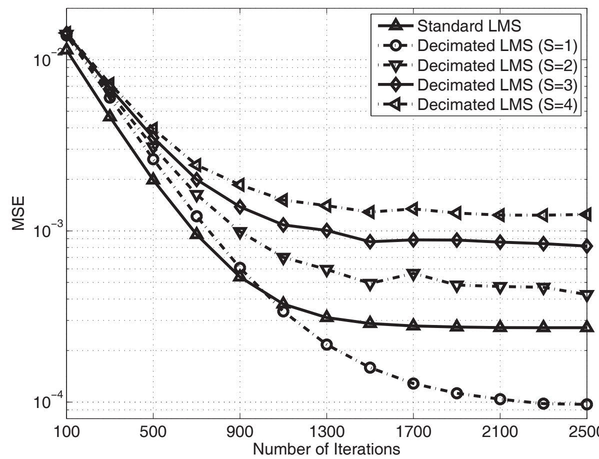

Fig. 3. Simulation example 2: Performance of the decimated LMS and the standard LMS for different sparsity levels.

$\rho_{ZAD} = 5 \times 10^{-4}$, $\mu = 0.05$. In the decimated LMS the parameter β in (6) is set to 0.1 when the signal to noise ratio (SNR) equals 10 dB, and it is set to 0.05 when SNR $= 20$ dB.

Fig. 2 shows the mean square errors (MSEs) of different estimation methods versus the length of the training sequence, i.e., the number of iterations. It can be seen that after a certain number of iterations the decimated LMS catches up with the standard LMS and it then displays a better performance despite it being less complex than the standard LMS. For both SNRs of 10 and 20 dB, the ZA-LMS has the best performance while the proposed ZAD-LMS method shows a better performance than the standard LMS for all training sequence lengths unlike the decimated LMS whose performance is worse than that of the LMS for small training sequences.

The second simulation example tests the effect of sparsity level on the performance of the decimated LMS. Since decimated LMS solves an l_1-norm minimization problem at the end of training to find an estimate of CIR, it is expected that its performance deteriorates with increasing the sparsity level of CIR. In this example, a CIR of length 64 is chosen. The sparsity basis is the DCT domain and the sparsity level of the CIR is varied from $S = 1$ to $S = 4$. The SNR is 10 dB, $\mu = 0.005$, and $\beta = 0.1$ in (6). Results are averaged over ten thousand simulation runs.

Fig. 3 depicts the MSE curves versus length of the training sequence. In this figure standard LMS is chosen as the point of reference and its performance is compared with decimated LMS for different sparsity levels. In order to allow for a fare comparison, the CIR's energy is kept the same which results in the same MSE values for standard LMS regardless of a specific sparsity level. It can be seen from Fig. 3 that the smaller the sparsity level of the signal the better is the performance of the decimated LMS as it is expected.

5. CONCLUSIONS

Two sparsity aware modifications of the standard LMS algorithm, which are the decimated LMS and ZAD-LMS, for frequency sparse channel estimation have been introduced motivated by the need of deriving channel estimation methods with lower complexity. The algorithms have been compared in terms of MSE to the standard LMS and the ZA-LMS. Simulations demonstrating the effectiveness of the proposed methods have been also shown.

6. REFERENCES

[1] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*, Prentice Hall, 1993.

[2] B. Widrow and S. D. Stearns, *Adaptive Signal Processing*, Prentice Hall, 1985.

[3] Y. Chen, Y. Gu, and A. O. Hero, "Sparse LMS for system identification," in *Proc. IEEE ICASSP*, Taipei, Taiwan, Apr. 2009, pp. 3125–3128.

[4] K. Shi, and X. Ma, "Transform domain LMS for sparse system identification," in *Proc. IEEE ICASSP*, Dallas, USA, Mar. 2010, pp. 3714–3717.

[5] J. Yang, and G. E. Sobelman, "Sparse LMS with segment zero attractors for adaptive estimation of sparse signals," in *Proc. IEEE APCCAS*, Kuala Lumpur, Malaysia, Dec. 2010, pp. 422–425.

[6] O. Taheri, and S. A. Vorobyov, "Sparse channel estimation with l_p-norm and reweighted l_1-norm penalized least mean squares," in *Proc. IEEE ICASSP*, Prague, Czech Republic, May 2011, pp. 2864–2867.

[7] D. Angelosante, J. A. Bazerque, and G. B. Giannakis, "Online adaptive estimation of sparse signals: where RLS meets the l_1-norm," *IEEE Trans. Signal Processing*, vol. 58, pp. 3436–3447, July 2010.

[8] N. Vaswani, "Kalman filtered compressed sensing," in *Proc. IEEE ICIP*, San Diego, USA, Oct. 2008, pp. 893–896.

[9] D. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, pp. 1289–1306, Apr. 2006.

[10] E. Candes and T. Tao, "Decoding by linear programming," *IEEE Trans. Inf. Theory*, vol. 51, pp. 4203–4215, Dec. 2005.

[11] E. J. Candes, and M. B. Wakin, "An introduction to compressive sampling," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 21–30, Mar. 2008.