Target Localization and Tracking in a Random Access Sensor Network

Kivanc Kerse, Fatemeh Fazel and Milica Stojanovic
Northeastern University
Email: kerse.k@husky.neu.edu, ffazel@ece.neu.edu, millitsa@ece.neu.edu

Abstract— **We consider tracking of multiple objects using a wireless sensor network where distributed nodes transmit to a fusion center using random access. During an initialization phase, targets are identified on a discrete set of locations using a sparse identification method. Tracking then proceeds to update the target locations and amplitudes explicitly, using a gradient algorithm to solve the underlying non-linear optimization problem. Updating continues at the pace dictated by the *average* sensing/transmission rate, which can be adjusted to suit an expected target velocity. By focusing explicitly on the target locations, as opposed to continuing with sparse identification over a quantized space whose size may be much greater than the number of targets, the goal is to reduce the computational complexity, improve the performance, and eliminate the spatial quantization effects.**

Index Terms— **Wireless sensor networks, target tracking, localization, sparse identification, gradient descent.**

I. Introduction

Wireless sensor networks have increasingly been considered for use in object detection, localization and tracking, e.g. [1]-[6]. Practical approximations to the otherwise intractable maximum-likelihood location estimation run the gamut from expectation-minimization and semi-definite relaxation techniques for computing the solution centrally [1]-[3] to techniques for distributed computation across the network nodes [4]. A particular trend in recent literature has been towards formulating localization as a sparse identification problem, where the understanding is that targets occupy only a few locations within a larger (or possibly much larger) grid of all candidate locations, e.g. [5]. By doing so, the intrinsically nonlinear localization problem is transformed into a linear one. Several algorithms have been proposed that capitalize on this fact and use Kalman filtering principles to address it [5], [6].

Our own work has focused on generic sparse field recovery within the setting of a random access sensor network [7]. In this setting, network nodes transmit at random instants in time, risking collision of data packets at the fusion center (FC), but sharing the limited available bandwidth in a simple manner. While occasional packet loss reduces the number of sensor readings available to the FC within a given interval of time, this loss can be counteracted by an appropriate increase in the sensing rate (packet transmission rate). In [7] we provide a closed-form relationship between the minimum per-node sensing rate, total available bandwidth, and the desired probability of field recovery.

In this paper, we combine the concept of random access sensing with the specific task of target tracking. In principle, this problem can be approached in two steps, where the FC first recovers an underlying sensing field (such as temperature, sound, or similar) from limited observations received randomly during a given time interval. In the second step, the recovered field is used by a detection/classification algorithm to identify the targets. An alternative approach is to consider the two steps simultaneously. We presently focus on such an approach, assuming that the FC has knowledge of target signatures, and uses it to localize the targets without engaging in the intermediate step of field recovery.

We decompose the target tracking problem into two phases: initial localization and subsequent tracking. Initial localization is performed using a standard technique for sparse system identification (e.g. mixed-norm minimization). The so-obtained estimates of target locations (and strengths) are used to initialize the tracking phase. In this phase, we dispense with the notion of sparse estimation, and instead pursue the moving target locations *explicitly* using a stochastic gradient descent to solve the underlying non-linear optimization problem.

The paper is organized as follows. In Sec.II we specify the system model. Sec.III is devoted to initial target localization within the random access framework, and Sec.IV is devoted to tracking. Numerical results are presented in Sec.V, and conclusions are summarized in Sec.VI.

II. System Model

We consider a certain geographical area populated by N sensors. The sensors measure the field generated by

M targets. This field at location z is modeled as

$$u(z) = \sum_{m=1}^{M} a_m g(z - c_m) \qquad (1)$$

where $g(z)$ is a known target signature, and a_m, c_m are the amplitude and location of target m, respectively.

Given a desired resolution d, the observation space is discretized into K points, such that a sensor placed at location z_k measures the signal $u_k = u(z_k)$, $k = 1, \ldots K$. If the resolution is fine enough, this signal can also be approximated as

$$u_k = \sum_{i=1}^{K} b_i \underbrace{g(z_k - z_i)}_{g_{k,i}} \qquad (2)$$

where

$$b_i = \begin{cases} a_m, \text{if } z_i = c_m \\ 0, \text{ otherwise} \end{cases} \qquad (3)$$

Collecting all the sensors' measurements into a vector, the full field can now be expressed as

$$\underbrace{\begin{bmatrix} u_1 \\ \vdots \\ u_K \end{bmatrix}}_{\mathbf{u}_F} = \underbrace{\begin{bmatrix} g_{1,1} & \cdots & g_{1,K} \\ \vdots & & \vdots \\ g_{K,1} & \cdots & g_{K,K} \end{bmatrix}}_{\mathbf{G}} \underbrace{\begin{bmatrix} b_1 \\ \vdots \\ b_K \end{bmatrix}}_{\mathbf{b}} \qquad (4)$$

Given the noisy measurements $v_k = u_k + w_k$, the problem of localization can be formulated as that of identifying the non-zero elements of the sparse vector $\mathbf{b}$ from the observations $\mathbf{v}_F = \mathbf{u}_F + \mathbf{w}_F$.

The FC, however, does not have all the measurements, but only a subset received from those sensors that were active during a given collection interval of duration T. This interval is chosen such that any motion can be neglected over its duration, i.e. the field can be approximated as constant during T. The subset of measurements received by the FC is given by $\mathbf{v}_R = \mathbf{R}\mathbf{v}_F$, where $\mathbf{R}$ is a random selection (reduction) matrix, obtained from a $K \times K$ identity matrix by removing those rows corresponding to locations with no sensors, inactive sensors, sensors whose packets collided, and sensors whose packets did not pass an error check (CRC) at the FC. The matrix $\mathbf{R}$ thus contains the effect of *communication* noise, while the *sensing* noise w_k is contained in the measurements v_k. The identity of contributing sensors is known to the FC, since each sensor tags its location onto the data packet that it transmits.

Combining the above expressions, we have that

$$\mathbf{v}_R = \mathbf{R}[\mathbf{G}\mathbf{b} + \mathbf{w}_F] = \mathbf{G}_R\mathbf{b} + \mathbf{w}_R \qquad (5)$$

This vector represents the measurements received by the FC during one collection interval.

III. Initial Localization

The initial collection interval is devoted to estimation of target locations and amplitudes, which are embodied in the sparse vector $\mathbf{b}$. The estimation problem can be formulated as mixed-norm minimization,

$$\hat{\mathbf{b}} = \arg\min_{\mathbf{b}} \{||\mathbf{v}_R - \mathbf{G}_R\mathbf{b}||_2^2 + \xi||\mathbf{b}||_1\} \qquad (6)$$

where $||\cdot||_l$ denotes the norm-l and ξ weighs the norm-1 contribution. The solution $\hat{\mathbf{b}}$ is readily obtained using linear programming, and this solution is used to initialize subsequent tracking.

Before we move on to tracking, a word is in order regarding the sensing rate needed to to make the initial estimate.

A. Choosing the sensing rate

The number of measurements received by the FC during T (size of $\mathbf{v}_R$) is a random variable, because the sensors transmit in a random access fashion. Following the treatment of [7], we model the arrival process of useful packets (those that did not collide, have passed the error check, and are not a repetition of the same packet) as a Poisson process $L(T)$ with the aggregate arrival rate

$$\lambda = \frac{N}{T} \underbrace{(1 - e^{-\lambda_1 T}) e^{-2N\lambda_1 T_p} (1 - P_E)}_{p} \qquad (7)$$

where λ_1 is the per-node sensing rate, T_p is the packet duration, and P_E is the probability of a packet being lost to communication noise.

While one cannot *guarantee* that the FC will receive sufficiently many packets (say L_s) during T, one can evaluate the *probability* P_s with which this will occur:

$$P_s = P\{L(T) \geq L_s\} = \sum_{L_s}^{\infty} \frac{(\lambda T)^l}{l!} e^{-\lambda T} \qquad (8)$$

System design now reduces to setting the value L_s (in accordance with the expected number of targets, sensing noise and the particular recovery algorithm) and requiring that the probability P_s be above a desired level P_s^*. This requirement implies the minimum necessary value of λT according to (8). For a given T (which is determined by the maximum expected target velocity), and a given packet duration T_p (which is determined by the packet size and the system bandwidth) a certain λT in turn implies a set of possible solutions for N and λ_1 according to (7). This procedure is illustrated in Fig.1. It is now up to the system designer to trade off between the number of sensors N and the per-node sensing rate λ_1. If the number of sensors is given, the per-node sensing rate follows immediately.

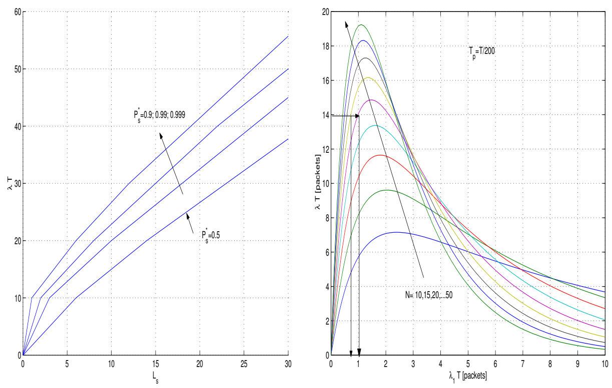

Fig. 1. A desired pair (L_s, P_s^*) implies the necessary λT (left), which in turn implies possible choices for N and λ_1 (right).

IV. Tracking

Once the initial estimate $\hat{\mathbf{b}} = \hat{\mathbf{b}}(0)$ is obtained, tracking can proceed to update $\hat{\mathbf{b}}(n)$, which can be done either in a batch mode (each new estimate computed independently from the previous ones), or in an adaptive fashion, e.g. via the RLS-LASSO algorithm [8]. Alternatively, tracking can focus *explicitly* on target locations $\hat{c}_m(n)$ and amplitudes $\hat{a}_m(n)$. We do so starting with the initial values $\hat{c}_m(0)$ and $\hat{a}_m(0)$ corresponding to the non-zero entries of $\hat{\mathbf{b}}(0)$, and continue to form

$$\hat{v}_k(n) = \sum_m \hat{a}_m(n) g(z_k - \hat{c}_m(n)), k \in \mathcal{R}_n \quad (9)$$

This signal estimate is formed for those location indices k where the measurements are available during the n-th collection interval (set $\mathcal{R}_n$), and the error is formed accordingly,

$$e_k(n) = v_k(n) - \hat{v}_k(n), k \in \mathcal{R}_n \quad (10)$$

Note that the number of measurements $L_n = |\mathcal{R}_n|$ is random, and will vary in each interval.

Taking all the measurements received during the n-th collection interval, we now form the instantaneous squared error

$$E(n) = \sum_{k \in \mathcal{R}_n} |e_k^2(n)| \quad (11)$$

whose partial derivatives yields the gradients

$$\frac{\partial E(n)}{\partial \hat{a}_m(n)} = -2 \sum_{k \in \mathcal{R}_n} e_k(n) g(z_k - \hat{c}_m(n)) \quad (12)$$

$$\frac{\partial E(n)}{\partial \hat{c}_m(n)} = 2 \sum_{k \in \mathcal{R}_n} e_k(n) \hat{a}_m(n) \dot{g}(z_k - \hat{c}_m(n)) \quad (13)$$

where $\dot{g}(z) = \frac{dg(z)}{dz} = \frac{\partial g(x+jy)}{\partial x} + j\frac{\partial g(x+jy)}{\partial y}$.

The above gradients exhibit nonlinear dependence on the location estimates, preventing one from obtaining a closed-form solution. A stochastic gradient algorithm can be employed instead to approach the solution in a recursive manner. Cast into the time-adaptive framework, the algorithm yields the updates

$$\hat{a}_m(n+1) = \hat{a}_m(n) + \mu \sum_{k \in \mathcal{R}_n} e_k(n) g(z_k - \hat{c}_m(n)) \quad (14)$$

$$\hat{c}_m(n+1) = \hat{c}_m(n) - \nu \underbrace{\sum_{k \in \mathcal{R}_n} e_k(n) \hat{a}_m(n) \dot{g}(z_k - \hat{c}_m(n))}_{\epsilon_m(n)} \quad (15)$$

where μ and ν are the a-priori set (positive) tracking constants. While this type of update is the simplest one, other types can be considered as well, e.g. an RLS-type amplitude update. In a practical application, the issue of location tracking may be more pressing than that of amplitude tracking, as target locations may change more rapidly than their amplitudes. To improve tracking in such conditions, a filtered location error gradient can be used:

$$\hat{c}_m(n+1) = \hat{c}_m(n) - \nu_1 \epsilon_m(n) - \nu_2 \sum_{i=1}^{n} \epsilon_m(n-i) \quad (16)$$

where ν_1, ν_2 are now two location tracking constants. Other filtering strategies are also possible, e.g. exponential weighting of the past values of $\epsilon_m(n)$.

Regardless of the specific update, it is important to note that the algorithm remains unchanged as the set of available observations (indices k at which the measurements are available) changes with time n. This fact has further implications on choosing the update interval, as it allows one to adjust T to suit a non-stationary process without altering the algorithm. Alternatively, instead of waiting for the end of the collection interval, updating can be performed with every new measurement received.

A. *Choosing the tracking constants*

A simplified single-target analysis offers an insight into the otherwise difficult optimization of the location-tracking loop. In the small-error regime, a linearized model of the single-order loop (15) leads to the closed-form expression for the mean-squared error (MSE) in the location estimate $\hat{c}(n)$:

$$\text{M}SE = \frac{\bar{\nu}}{2 - \bar{\nu}} \cdot \frac{1}{SNR} + \left(\frac{vT}{\bar{\nu}}\right)^2 \quad (17)$$

where $\bar{\nu} = \rho A^2 \nu$ is the normalized tracking constant, $SNR = \rho A^2 / \sigma_w^2$ is the effective SNR (σ_w^2 is the variance of the sensing noise), v is the target velocity, and $\rho = \sum_{k \in \mathcal{R}_n} \dot{g}^2(z_k - c)$. Assuming presence of a target at all times, and sufficient spatial resolution,

$\rho \sim (p/\Delta)\int_{-\infty}^{+\infty}[\dot{g}(z)]^2 dz$, where Δ is the separation between sensors and p is the probability that a sensor is contributing a useful measurement. This simplified analysis implies the stability condition $\bar{\nu} < 2$, as well as the optimal value $\bar{\nu}_{opt}$ for which the MSE is minimal.

V. Performance Results

To illustrate the system performance, we consider a network of N=10 sensors monitoring a field generated by M=3 targets. For clarity of illustration (but without loss of generality) we focus on one-dimensional geometry, where targets move along a track in different directions and at different speeds. The targets have exponentially decaying signatures, $g(z) = e^{-\alpha|z|}$, where α [m^{-1}] specifies the decay rate. The resulting field is shown in Fig.2 for two cases with different decay rates.

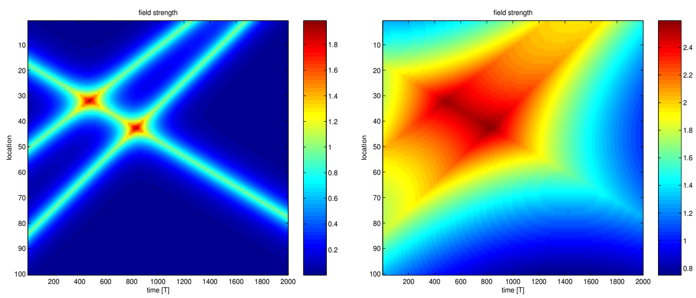

Fig. 2. Sensing field as it evolves across one track (vertical line) over time. There are there targets with unit amplitudes and exponential signature decay rates of 0.2 (left) and 0.02 (right) per unit distance. Assuming a resolution of 1 m in space and 0.1 s in time, the targets start at mid-points of the 100 m track and move at velocities 0.3 m/s, -0.4 m/s and -0.5 m/s during a total observation window of 200 s. While the targets on the left can be identified by a naked eye, those on the right cannot (slow decay rate causes smearing of signatures).

The sensors are placed uniformly along the track,[1] and sample the field at random instants in time. Setting T=0.1 s, $T_p = T/200$, $\lambda_1 = 2/T$, and $P_E < 0.01$, the probability that a sensor will contribute a useful measurement during T is $p \approx 0.7$. The resulting measurements that are conveyed to the FC are shown in Fig.3.

The performance of target tracking in the absence of noise is shown in Fig.4. In all of the cases, ν_1 is taken at least 10 times below the single-target optimum at 0.5 m/s, $\nu_2 = \nu_1/10$, and $\mu = 0.001$. Perfect initial localization is assumed. Shown along with explicit tracking (16) are the location estimates obtained by the matching pursuit algorithm [9] applied in batch mode.

Explicit tracking clearly succeeds (its estimate is indistinguishable from the true trajectory) not only in following the targets while they are within the observation window, but also as they leave it (one at around 1200 T

[1]For the present study, sensors are fixed (not moving).

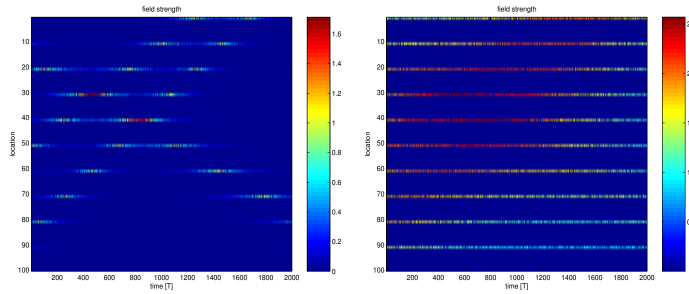

Fig. 3. The field of Fig.2 as seen by the fusion center.

and another at around 1600 T). In contrast, sparse estimation based on matching pursuit fails completely. There are several reasons for its failure. One is the lack of observations (size of the vector $\mathbf{v}_R$) which is caused by having too few sensors (insufficient spatial resolution), low per-node sensing rate, or too many collisions. The other the lack of orthogonality between the columns of the matrix $\mathbf{G}_R$, which occurs with slowly decaying target signatures. The performance improves if these parameters change, but only to a certain extent. Fig.5 shows a hypothetical best case in which there are $N = K$=100 sensors (resolution 1 m), all the sensors transmit in every collection interval, and there are no collisions.

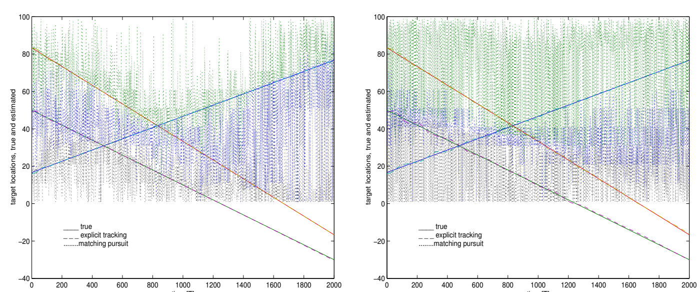

Fig. 4. True and estimated target locations vs. time. Decay rate is 0.2 per unit distance (left) or 0.02 (right). No noise, N=10 sensors contributing with probability 0.7.

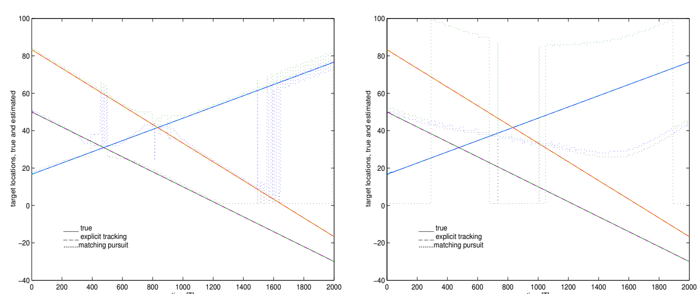

Fig. 5. True and estimated target locations vs. time. Decay rate is 0.2 per unit distance (left) or 0.02 (right). No noise, N=100 sensors contributing with probability 1.

Next we address the effect of sensing noise, focusing on the case with slowly decaying target signatures (α=0.02 m^{-1}). Fig.6 illustrates the performance obtained at the effective SNR of -20 dB. While there is clearly some degradation due to noise, tracking remains functional. Fig.7 summarizes the results. Squared location error, measured for each target in every collection interval, is averaged over all targets and all collection intervals within the total observation window. The process is repeated over 100 independent realizations of the noise, random sensor activation and collisions. This result shows that explicit tracking can indeed maintain good performance in the presence of noise, that it can do so for a range of target signatures (resolvable or smeared) and that it can tolerate trajectories that take targets outside of the area populated by the sensors.

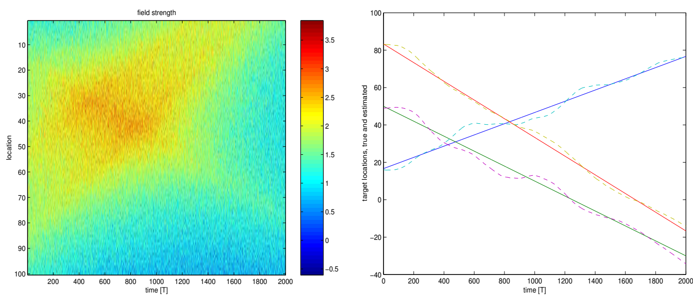

Fig. 6. Target tracking in the presence of noise. Decay rate is 0.02 per unit distance. Noise at effective SNR of -20 dB, N=10 sensors contributing with probability 0.7.

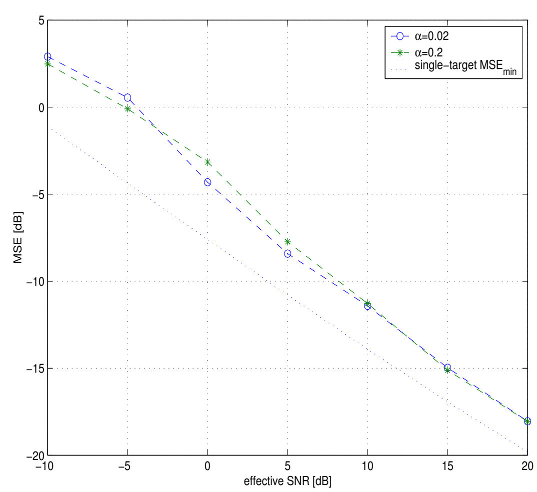

Fig. 7. Summary of numerical results. N=10 sensors, p=0.7.

VI. Conclusion

We addressed the problem of target tracking in the context of a random access sensor network. Tracking is initialized by locating the targets in discrete space using a standard sparse identification technique based on convex optimization or a greedy method. From there on, sparse estimation is abandoned in favor of explicit location and amplitude tracking. By doing so, the goal is to reduce the computational complexity and improve the performance in cases where few targets are covering a large space. For example, three targets moving along a 1 km stretch represented with 1 m resolution require identification of three out of 1000 elements of an unknown vector, while explicit tracking focuses on the three unknown locations only. The estimated locations are continuous-valued, i.e. they do not depend on the chosen resolution (grid size, quantization). Explicit tracking also tolerates situations in which the targets move outside of the pre-defined window (e.g. the 1 km stretch), while sparse estimation techniques require this pre-defined window to increase, thus further increasing complexity.

Future research will focus on sensitivity analysis, and on addressing system-level feedback control whereby the FC can instruct the sensors to (a) increase their average transmission rate (so as to keep up with an accelerating target) or decrease it (so as to save transmission energy during uneventful periods), and (b) move towards areas of more interest (mobile sensors).

References

[1] X. Sheng and Y. Hu, Maximum likelihood multiple source localization using acoustic energy measurements with wireless sensor networks, IEEE Trans. Sig. Proc., vol.53, No.1, Jan. 2005.

[2] W. Meng, W. Xiao, and L. Xie, An efficient EM algorithm for energy-based multi-source localization in wireless sensor networks, IEEE Trans. Instrumentation and Measurement, vol.60, No.3, pp.10171027, March 2011.

[3] M. Chen, D. Zhi, and S. Dasgupta, A semidefinite programming approach to source localization in wireless sensor networks, IEEE Sig. Proc. Letters, vol.15, pp.253256, 2008. No.2, pp.1729, March 2002.

[4] D. Blatt and A.O. Hero, Energy-based sensor network source localization via projection onto convex sets, IEEE Trans. Sig. Proc., vol.54, No.9, pp.36143619, What 2006.

[5] S. Farahmand, G.B. Giannakis, G. Leus, and Z. Tian, Tracking target signal strengths on a grid using sparsity, 2011, http://arxiv.org/abs/1104.5288.

[6] E. Masazade, M. Fardad, and P.K. Varshney, Sparsity-promoting extended Kalman filtering for target tracking in wireless sensor networks, IEEE Sig. Proc. Letters, vol.19, No.12, pp. 845848, Dec. 2012.

[7] F. Fazel, M. Fazel, and M. Stojanovic, Random access compressed sensing over fading and noisy communication channel, IEEE Trans. Wireless Commun., vol.12, No.5, pp.21142125, May 2013.

[8] D. Angelosante and G. Giannakis, "RLS-weighted LASSO for adaptive estimation of sparse signals," in Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) 2009.

[9] S. Mallat and Z.Zhang, "Matching pursuits with time-frequency dictionaries," IEEE trans. Sig. Proc., vol.41, No.12, pp.3397,3415, Dec. 1993.