Robust RLS in the Presence of Correlated Noise Using Outlier Sparsity

Shahrokh Farahmand and Georgios B. Giannakis

***Abstract*—Relative to batch alternatives, the recursive least-squares (RLS) algorithm has well-appreciated merits of reduced complexity and storage requirements for online processing of stationary signals, and also for tracking slowly-varying nonstationary signals. However, RLS is challenged when in addition to noise, outliers are also present in the data due to, e.g., impulsive disturbances. Existing robust RLS approaches are resilient to outliers, but require the nominal noise to be white—an assumption that may not hold in e.g., sensor networks where neighboring sensors are affected by correlated ambient noise. Prewhitening with the known noise covariance is not a viable option because it spreads the outliers to noncontaminated measurements, which leads to inefficient utilization of the available data and unsatisfactory performance. In this correspondence, a robust RLS algorithm is developed capable of handling outliers and correlated noise simultaneously. In the proposed method, outliers are treated as nuisance variables and estimated jointly with the wanted parameters. Identifiability is ensured by exploiting the sparsity of outliers, which is effected via regularizing the least-squares (LS) criterion with the ℓ_1-norm of the outlier vectors. This leads to an optimization problem whose solution yields the robust RLS estimates. For low-complexity real-time operation, a suboptimal online algorithm is proposed, which entails closed-form updates per time step in the spirit of RLS. Simulations demonstrate the effectiveness and improved performance of the proposed method in comparison with the nonrobust RLS, and its state-of-the-art robust renditions.**

***Index Terms*—Correlated noise, outlier resilience, recursive least-squares, robust statistics, sparsity.**

I. Introduction

Online processing with reduced complexity and storage requirements relative to batch alternatives, as well as adaptability to slow parameter variations, are the main reasons behind the popularity of the recursive least-squares (RLS) algorithm for adaptive estimation of linear regression models. Despite its success in various applications, RLS is challenged by the presence of *outliers*, that is measurement or noise samples not adhering to a prespecified nominal model. One major source of outliers is impulsive noise, which appears for example as "double-talk" in an echo cancelation setting [13]. In another application, which motivates the present contribution, outliers may arise with low-cost unreliable sensors deployed for estimating the coefficients of a linear regression model involving also (possibly correlated) ambient noise. Outlier-resilient RLS approaches are imperative in such applications; see, e.g., [10].

Robust RLS schemes have been reported in [3], [5], [12], [13], and [16]. Huber's M-estimation criterion was considered in [12], Hampel's three-part redescending M-estimate was advocated in [16], and a modified Huber cost was put forth in [5]. These approaches improve upon the nonrobust RLS when outliers are present, but they also have limitations: i) some entail costs that are not convex [5], thus providing no guarantee for achieving the global optimum; ii) the optimization problem to be solved per time step has dimensionality (and hence computations and storage) growing linearly with time; and iii) correlated nominal noise of known covariance cannot be handled, which is critical for estimation using a network of distributed sensors because ambient noise is typically dependent across neighboring sensors.

To mitigate the "curse of dimensionality," [5] provides suboptimal online recursions in the spirit of RLS, while [3] and [13] limit the effect of each new datum on the estimate, thus also ensuring low-complexity RLS-like iterations. However, all existing approaches cannot handle correlated ambient noise of known covariance. It is worth stressing that prewhitening is not recommended because it spreads the outliers to noncontaminated measurements rendering outlier-resilient estimation even more challenging.

In this correspondence, outliers are treated as nuisance variables that are jointly estimated with the wanted regression coefficients. The identifiability challenge arising due to the extra unknowns is addressed by exploiting the outlier sparsity. The latter is effected by regularizing the LS cost with the ℓ_1-norm of the outlier vectors. Consequently, one arrives at an optimization problem that can identify outliers while at the same time it can accommodate correlated ambient noise. Since dimensionality of this optimization problem also grows with time, its solution is utilized at the initialization stage and provides an offline benchmark. To cope with dimensionality, an online variant is also developed, which enjoys low-complexity and closed-form updates in the spirit of RLS. Initializing real-time updates with the offline benchmark, the novel online robust RLS (OR-RLS) algorithm is compared against RLS and the robust RLS schemes in [3] and [13] via simulations, which demonstrate its efficiency and improved performance.

II. Problem Statement and Preliminaries

Consider the *outlier-aware* linear regression model considered originally in [7], where (generally vector) measurements $\mathbf{y}_k \in \mathbb{R}^{D_y}$ involving the unknown vector $\mathbf{x}_k \in \mathbb{R}^{D_x}$ become available sequentially in time (indexed by k), in the presence of (possibly correlated) nominal noise $\mathbf{v}_k$; that is

$$\mathbf{y}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k + \mathbf{o}_k \tag{1}$$

where $\mathbf{H}_k$ denotes a known regression matrix, $\mathbf{o}_k$ models the outliers arising from e.g., impulsive noise, and $\mathbf{v}_k$ is zero-mean with known covariance matrix $\mathbf{R}_k$, uncorrelated with $\mathbf{v}_{1:k-1} := [\mathbf{v}_1^T, \mathbf{v}_2^T, \ldots, \mathbf{v}_{k-1}^T]^T$, but possibly correlated across its own entries. The wanted vector $\mathbf{x}_k$ is either constant, or, varies slowly with k. The goal is to estimate $\mathbf{x}_k$ and $\mathbf{o}_k$, given the measurements $\mathbf{y}_{1:k} := [\mathbf{y}_1^T, \mathbf{y}_2^T, \ldots, \mathbf{y}_k^T]^T$. To motivate the vector model in (1), consider a sensor network with each sensor measuring distances (in rectangular coordinates) between itself and the target that is desired to localize and track. The sensor measurements are synchronized by the clock of an access point (AP), and are corrupted by outliers as well as nominal noise that is correlated across neighboring sensors.

Had $\mathbf{o}_k$ been equal to zero or *known* for that matter, one could apply the RLS algorithm on the outlier compensated measurements $\mathbf{y}_k - \mathbf{o}_k$. Therefore, at time step k, one would solve [11, p. 225]

$$\begin{aligned}\hat{\mathbf{x}}_k &:= \arg\min_{\mathbf{x}_k} \mathcal{J}(\mathbf{x}_k)\\ &:= \arg\min_{\mathbf{x}_k} \frac{1}{2}\sum_{i=1}^{k} \beta^{k-i}\|\mathbf{y}_i - \mathbf{H}_i\mathbf{x}_k - \mathbf{o}_i\|^2_{\mathbf{R}_i^{-1}}\end{aligned} \tag{2}$$

Manuscript received June 16, 2011; revised December 26, 2011; accepted February 16, 2012. Date of publication March 05, 2012; date of current version May 11, 2012. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Philippe Ciblat.

This work was supported by the AFOSR MURI Grant FA9550-10-1-0567.

The authors are with the Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN 55455 USA (e-mail: shahrokh@umn.edu; georgios@umn.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSP.2012.2189766

where $0 < \beta \leq 1$ is a preselected forgetting factor, and $\|\mathbf{x}\|_{\mathbf{R}}^2 := \mathbf{x}^T\mathbf{R}\mathbf{x}$. Equating to zero the gradient of $\mathcal{J}(\mathbf{x}_k)$ in (2), yields

$$\left(\sum_{i=1}^{k}\beta^{k-i}\mathbf{H}_i^T\mathbf{R}_i^{-1}\mathbf{H}_i\right)\hat{\mathbf{x}}_k = \left(\sum_{i=1}^{k}\beta^{k-i}\mathbf{H}_i^T\mathbf{R}_i^{-1}(\mathbf{y}_i - \mathbf{o}_i)\right)$$

or in a compact matrix-vector notation as $\mathbf{\Phi}_k^{-1}\hat{\mathbf{x}}_k = \boldsymbol{\phi}_k$, with obvious definitions for $\boldsymbol{\phi}_k$ and $\mathbf{\Phi}_k$ assumed invertible. The latter can be solved online using the RLS recursions (see, e.g., [11])

$$\begin{aligned}\hat{\mathbf{x}}_k &= \hat{\mathbf{x}}_{k-1} + \mathbf{W}_k(\mathbf{y}_k - \mathbf{o}_k - \mathbf{H}_k\hat{\mathbf{x}}_{k-1}) \\ \mathbf{W}_k &= \beta^{-1}\mathbf{\Phi}_{k-1}\mathbf{H}_k^T(\mathbf{H}_k\beta^{-1}\mathbf{\Phi}_{k-1}\mathbf{H}_k^T + \mathbf{R}_k)^{-1} \\ \mathbf{\Phi}_k &= \beta^{-1}\mathbf{\Phi}_{k-1} - \mathbf{W}_k\mathbf{H}_k\beta^{-1}\mathbf{\Phi}_{k-1}.\end{aligned} \tag{3}$$

The challenge with $\mathbf{o}_k$ being *unknown* in addition to $\mathbf{x}_k$ is that (1) is under-determined, which gives rise to identifiability issues if $\mathcal{J}$ in (2) is to be *jointly* optimized over $\mathbf{x}_k$ and $\mathbf{o}_{1:k} := [\mathbf{o}_1^T \ldots \mathbf{o}_k^T]^T$. Indeed, e.g., $\mathbf{o}_k = \mathbf{y}_k$ and $\mathbf{x}_k = \mathbf{0}$ is one choice of the unknown vectors always minimizing $\mathcal{J}$ in (2).

The ensuing section develops algorithms allowing joint estimation of $\mathbf{x}_k$ and $\mathbf{o}_{1:k}$ (or $\mathbf{o}_k$ only), by exploiting the attribute of sparsity present in the outlier vectors.

III. Outlier-Sparsity-Aware Robust RLS

To ensure uniqueness in the joint optimization over $\mathbf{o}_{1:k}$ and $\mathbf{x}_k$, consider regularizing the cost in (2). Among candidate regularizers, popular ones include the ℓ_p-norm of outlier vectors with $p = 0, 1, 2$. A key observation guiding the selection of p, is that outliers arising from sources such as impulsive noise are sparse. The degree of sparsity is quantified by the ℓ_0-(pseudo)norm of $\mathbf{o}_i$, which equals the number of its nonzero entries. Unfortunately, the ℓ_0-norm makes the optimization problem nonconvex, and in fact NP-hard to solve. This motivates adopting the closest convex approximation of ℓ_0, namely the ℓ_1-norm, which is known to approach (and under certain conditions on the regressors coincide with) the solution obtained with the ℓ_0-norm [4]. Different from [1], [4] that rely on the ℓ_1-norm of sparse vectors $\mathbf{x}_k$, the novelty here is adoption of the ℓ_1-norm of sparse outlier vectors for robust RLS processing.

Adopting this regularization of $\mathcal{J}$ in (2), yields

$$[\hat{\mathbf{x}}_k, \hat{\mathbf{o}}_{1:k}] = \arg\min_{\mathbf{x}_k,\mathbf{o}_{1:k}} \sum_{i=1}^{k}\beta^{k-i}\left[\frac{1}{2}\|\mathbf{y}_i - \mathbf{H}_i\mathbf{x}_k - \mathbf{o}_i\|_{\mathbf{R}_i^{-1}}^2 + \lambda_i\|\mathbf{o}_i\|_1\right]. \tag{4}$$

Besides being convex, (4) is nicely linked with Huber's M-estimates if the nominal noise $\mathbf{v}_i$ is white [8]. Indeed, with $\mathbf{R}_i = \mathbf{I}\ \forall\, i$, solving (4) amounts to solving [8]

$$\hat{\mathbf{x}}_k = \arg\min_{\mathbf{x}_k}\sum_{i=1}^{k}\sum_{d=1}^{D_y}\beta^{k-i}\rho_i(y_{i,d} - \mathbf{h}_{i,d}^T\mathbf{x}_k) \tag{5}$$

where $y_{i,d}(\mathbf{h}_{i,d}^T)$ denotes the dth entry (row) of $\mathbf{y}_i$ (respectively, $\mathbf{H}_i$), and ρ_i is Huber's function given by

$$\rho_i(x) := \begin{cases}\frac{x^2}{2}, & |x| \leq \lambda_i \\ |x|\lambda_i - \frac{\lambda_i^2}{2}, & |x| > \lambda_i.\end{cases}$$

While solving (4) recovers an estimate of $\mathbf{x}_k$, its practicality is progressively compromised with time because the number of variables to be estimated increases with k. In addition, (4) cannot be solved recursively, and at each time instant the entire optimization should be carried out anew. This shortcoming affects Huber M-estimates too, and similar limitations have been identified also by [5] and [16]. For this reason, the offline benchmark solver of (4) will be used as a "warm-start" to initialize online methods developed in Section III-A.

A. Online Robust RLS (OR-RLS)

Since it is impractical to re-estimate past and current outliers per time instant k, the main idea in this section is to estimate only the most recent outliers. Consequently, the proposed OR-RLS proceeds in two steps per k. Given $\hat{\mathbf{x}}_{k-1}$ from $k-1$, the current $\mathbf{o}_k$ is estimated in the first step via

$$\hat{\mathbf{o}}_k = \arg\min_{\mathbf{o}_k}\left[\frac{1}{2}\|\mathbf{y}_k - \mathbf{H}_k\hat{\mathbf{x}}_{k-1} - \mathbf{o}_k\|_{\mathbf{R}_k^{-1}}^2 + \lambda_k\|\mathbf{o}_k\|_1\right]. \tag{6}$$

With $\hat{\mathbf{o}}_k$ available, ordinary RLS is applied to the outlier-free measurements $\mathbf{y}_k - \hat{\mathbf{o}}_k$ [cf. (3)] in the second step, which incurs complexity comparable to ordinary RLS. Clearly, (6) solves a problem of fixed dimension D_y per time step; hence, it offers a practical alternative to the offline benchmark in (4).

If $\mathbf{R}_k$ is *diagonal*, denote it as $\mathrm{diag}(r_{k,11}, r_{k,22}, \ldots, r_{k,D_yD_y})$, then (6) decouples into scalar subproblems across entries of $\mathbf{o}_k$ as

$$\hat{o}_{k,d} = \arg\min_{o_{k,d}}\left[\frac{(y_{k,d} - \mathbf{h}_{k,d}^T\hat{\mathbf{x}}_{k-1} - o_{k,d})^2}{2r_{k,dd}} + \lambda_k|o_{k,d}|\right], \quad d = 1, \ldots, D_y.$$

The solution to these subproblems is available in closed form using the so termed least-absolute shrinkage and selection operator (Lasso); see, e.g., [9]. In the present context, this closed-form solution is given by the so-termed soft-thresholding operator as

$$\hat{o}_{k,d} = \max\left\{|y_{k,d} - \mathbf{h}_{k,d}^T\hat{\mathbf{x}}_{k-1}| - \lambda_k r_{k,dd}, 0\right\} \times \mathrm{sign}(y_{k,d} - \mathbf{h}_{k,d}^T\hat{\mathbf{x}}_{k-1}), \quad d = 1, \ldots, D_y. \tag{7}$$

Thus, OR-RLS with white nominal noise $\mathbf{v}_k$ amounts to simple noniterative closed-form evaluations per time step k. However, when $\mathbf{v}_k$ is colored, (6) can no longer be solved in closed form.

If $\mathbf{R}_k$ is *nondiagonal*, general-purpose convex solvers such as SeDuMi can be utilized to solve (6) with complexity $O(D_y^{3.5})$ [14]. In Sections III-B–III-D, two methods based on coordinate descent (CD), and the alternating direction method of multipliers (AD-MoM) are developed to solve (6) iteratively. While both are asymptotically convergent, it suffices to run only a few iterations in practice, which reduces complexity of general solvers and parallels the low-complexity of OR-RLS when $\mathbf{v}_k$ is white.

B. CD Based OR-RLS

To iteratively solve (6) with respect to $\mathbf{o}_k$, consider initializing with $\mathbf{o}_k^{(0)} = \mathbf{0}$. Then, for $j = 1, \ldots, J$, vector $\mathbf{o}_k^{(j)}$ is updated one entry at a time using CD. Specifically, the dth entry at iteration j is updated as

$$o_{k,d}^{(j)} := \arg\min_{o_{k,d}}\frac{1}{2}\left\|\mathbf{y}_k - \mathbf{H}_k\hat{\mathbf{x}}_{k-1} - \begin{bmatrix}\mathbf{o}_{k,1:d-1}^{(j)} \\ o_{k,d} \\ \mathbf{o}_{k,d+1:D_y}^{(j-1)}\end{bmatrix}\right\|_{\mathbf{R}_k^{-1}}^2 + \lambda_k|o_{k,d}|. \tag{8}$$

Problem (8) can be alternatively written as a scalar Lasso one, that is

$$o_{k,d}^{(j)} := \arg\min_{o_{k,d}}\frac{1}{2}\left(o_{k,d} - \gamma_{k,d}^{(j)}\right)^2 + \lambda_{k,d}|o_{k,d}| \tag{9}$$

TABLE I
OR-RLS ALGORITHM

```
Initialization. Collect measurements $\mathbf{y}_k$ for $k = 1, \ldots, k_0$. Solve (11) to obtain $\hat{\mathbf{x}}_{k_0}$.
              Use either of the two methods in [6] to determine the best $\lambda_k = \lambda_y,\ \forall\ k$
Repeat for $k \geq k_0 + 1$
    Solve (6) to obtain $\hat{\mathbf{o}}_k$. Use CD iterations in (9) or AD-MoM iterations in (10).
    Run RLS in (3) for the outlier compensated measurements $\mathbf{y}_k - \hat{\mathbf{o}}_k$ to obtain $\hat{\mathbf{x}}_k$.
End for
```

where

$$\gamma_{k,d}^{(j)} := \frac{1}{\tilde{r}_{k,d,d}}[\alpha_{k,d} - \sum_{i=1}^{d-1} \tilde{r}_{k,i,d} o_{k,i}^{(j)} - \sum_{i=d+1}^{D_y} \tilde{r}_{k,i,d} o_{k,i}^{(j-1)}]$$
$$\boldsymbol{\alpha}_k := \mathbf{R}_k^{-1}(\mathbf{y}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k-1}), \qquad \lambda_{k,d} := \frac{\lambda_k}{\tilde{r}\ _{k,d,d}}$$

with $\tilde{r}_{k,d,d'} := [\mathbf{R}_k^{-1}]_{d,d'}$ denoting the (d, d') entry of $\mathbf{R}_k^{-1}$. The solution to (9) is given by (cf. (7))

$$o_{k,d}^{(j)} = \max\left\{\left|\gamma_{k,d}^{(j)}\right| - \lambda_{k,d}, 0\right\} \text{ sign } \left(\gamma_{k,d}^{(j)}\right).$$

After J iterations, one sets $\hat{\mathbf{o}}_k = \mathbf{o}_k^{(J)}$. Global convergence of the iterates $o_k^{(J)}$ to the solution of (6) is guaranteed as $J \to \infty$ from the results in [15]. In practice however, a fixed small J suffices.

C. AD-MoM Based OR-RLS

To decouple the nondifferentiable term of the cost in (6) from the differentiable one, consider introducing the auxiliary variable $\mathbf{c}_k$, and re-writing (6) in constrained form as

$$[\hat{\mathbf{o}}_k, \hat{\mathbf{c}}_k] = \arg \min_{\mathbf{o}_k, \mathbf{c}_k} \left[\frac{1}{2}\|\mathbf{y}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k-1} - \mathbf{o}_k\|_{\mathbf{R}_k^{-1}}^2 + \lambda_k \|\mathbf{c}_k\|_1\right]$$
$$\text{subject to} \quad \mathbf{o}_k = \mathbf{c}_k.$$

The augmented Lagrangian thus becomes

$$\begin{aligned}\mathcal{L}(\mathbf{o}_k, \mathbf{c}_k, \boldsymbol{\mu}) =& \frac{1}{2}\|\mathbf{y}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k-1} - \mathbf{o}_k\|_{\mathbf{R}_k^{-1}}^2 + \lambda_k \|\mathbf{c}_k\|_1 \\ &+ \boldsymbol{\mu}^T(\mathbf{o}_k - \mathbf{c}_k) + \frac{\kappa}{2}\|\mathbf{o}_k - \mathbf{c}_k\|_2^2\end{aligned}$$

where $\boldsymbol{\mu}$ is the Lagrange multiplier, and κ is a positive constant controlling the effect of the quadratic term introduced to ensure strict convexity of the cost. After initializing with $\mathbf{c}_k^{(0)} = \boldsymbol{\mu}^{(0)} = \mathbf{0}$, AD-MoM recursions per iteration j (see, e.g., [2]), in the present context amount to

$$\begin{aligned}\mathbf{o}_k^{(j)} =& (\mathbf{R}_k^{-1} + \kappa \mathbf{I})^{-1} \\ &\times \left(\mathbf{R}_k^{-1}(\mathbf{y}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k-1}) + \kappa \mathbf{c}_k^{(j-1)} + \boldsymbol{\mu}^{(j-1)}\right) \quad \text{(10a)}\\ c_{k,d}^{(j)} =& \max\left\{\left|o_{k,d}^{(j)} + \frac{\mu_d^{(j)}}{\kappa}\right| - \frac{\lambda_k}{\kappa}, 0\right\} \text{sign}\left(o_{k,d}^{(j)} + \frac{\mu_d^{(j)}}{\kappa}\right), \\ & \qquad d = 1, \ldots, D_y \quad \text{(10b)}\\ \boldsymbol{\mu}^{(j)} =& \boldsymbol{\mu}^{(j-1)} + \kappa(\mathbf{o}_k^{(j)} - \mathbf{c}_k^{(j)}). \quad \text{(10c)}\end{aligned}$$

After J iterations, set $\hat{\mathbf{o}}_k = \mathbf{o}_k^{(J)}$. Global convergence of the iterates $o_k^{(J)}$ to the solution of (6) is guaranteed as $J \to \infty$ from the results in [2, p. 256]. As with CD, a fixed small J suffices in practice. Note also that the matrix $\mathbf{R}_k^{-1}$ needs to be computed only once, and used throughout the iterations.

D. Parameter Selection

Successful performance of OR-RLS hinges on proper selection of the sparsity-enforcing coefficients λ_k in (8) and (10). Too large a λ_k brings OR-RLS back to the ordinary RLS, while too small a λ_k creates many spurious outliers and degrades OR-RLS performance. Suppose that λ_k is constant with respect to k; that is, let $\lambda_k = \lambda_y$ for all $k \geq 1$.

Judicious tuning of λ_y has been investigated for robust *batch* linear regression estimators, and two criteria to optimally select λ_y can be found in [6] and [8]. However, none of these algorithms can be run *online*. As a remedy, it is recommended to select λ_y during the initialization (batch) phase of OR-RLS, and retain its value for all future time instants. OR-RLS initialization is carried out by solving the offline benchmark for $k = 1$ up to $k = k_0$ using SeDuMi; that is

$$[\hat{\mathbf{x}}_{k_0}, \hat{\mathbf{o}}_{1:k_0}] = \arg \min_{\mathbf{x}_{k_0}, \mathbf{o}_{1:k_0}} \sum_{i=1}^{k_0} \beta^{k_0 - i}\left[\frac{1}{2}\|\mathbf{y}_i - \mathbf{H}_i \mathbf{x}_{k_0} - \mathbf{o}_i\|_{\mathbf{R}_i^{-1}}^2 + \lambda_y \|\mathbf{o}_i\|_1\right]. \quad (11)$$

At time k_0, either one of the two criteria in [6] is applied to (11) to find the "best" λ_y. Using this λ_y, (11) is solved to obtain the initial $\hat{\mathbf{x}}_{k_0}$. Then, one proceeds to time $k_0 + 1$ and runs the OR-RLS algorithm as detailed in the previous subsections. Table I summarizes the OR-RLS algorithm.

Remark 1 (Complexity of OR-RLS): This remark quantifies the complexity order of OR-RLS per time index k. Starting from (3), the second subequation requires inversion of a $D_y \times D_y$ matrix, which incurs complexity $\mathcal{O}(D_y^3)$ once per time index k; while the third subequation involves matrix multiplications at complexity $\mathcal{O}(D_x^2 D_y)$. Hence, the complexity of the nonrobust RLS for the vector-matrix model (1) is $\mathcal{O}(D_y^3 + D_x^2 D_y)$. For OR-RLS, one needs to solve (6) per time step also. Utilizing SeDuMi to solve (6) incurs complexity $\mathcal{O}(D_y^{3.5} + D_x D_y)$, where the second term corresponds to evaluating $\mathbf{H}_k \hat{\mathbf{x}}_{k-1}$. Consider now CD or AD-MoM iterations. Per time index k, CD iterations must compute $\boldsymbol{\alpha}_k$, which incurs complexity $\mathcal{O}(D_y^3)$ for matrix inversion, and $\mathcal{O}(D_x D_y)$ for matrix-vector multiplication. In each of the J CD iterations, one needs to evaluate D_y times $\gamma_{k,d}^{(j)}$, which incurs overall complexity $\mathcal{O}(J D_y^2)$. Thus, the overall complexity of optimizing (6) via CD iterations is $\mathcal{O}(D_y^3 + D_x D_y + J D_y^2)$. In as much as J is selected on the order of $\mathcal{O}(D_y)$ or smaller, CD iterations do not increase the complexity order of the nonrobust RLS in (3). Furthermore, its complexity order will be smaller than that of SeDuMi. With regards to AD-MoM, two $D_y \times D_y$ matrices must be inverted once per time step k at complexity $\mathcal{O}(D_y^3)$. The matrix-vector multiplication $\mathbf{H}_k \hat{\mathbf{x}}_{k-1}$ is also performed once per time index at complexity $\mathcal{O}(D_x D_y)$. Matrix-vector multiplications in (10a) should be carried out J times, which incurs complexity $\mathcal{O}(J D_y^2)$. Thus, the overall complexity of optimizing (6) via AD-MoM iterations is $\mathcal{O}(D_y^3 + D_x D_y + J D_y^2)$. Again, if J is on the order of $\mathcal{O}(D_y)$ or smaller, AD-MoM iterations do not incur higher complexity than the nonrobust RLS in (3).

Remark 2 (Complexity of Alternative Robust RLS Methods): For comparison purposes with OR-RLS, this remark discusses the complexity order of the RLS schemes in [3], [5], [13], and [16]. Focusing for simplicity on [13] which entails nonrobust RLS recursions, consider correlated measurements arriving in vectors of dimension D_y. To decorrelate, prewhitening with the square root matrix incurs complexity $\mathcal{O}(D_y^3)$. For each new datum extracted after prewhitening, vector $\mathbf{k}_i$ in [13], (5), involves vector-matrix products at complexity $\mathcal{O}(D_x^2)$. With D_y such data per time step, the overall complexity grows to $\mathcal{O}(D_x^2 D_y)$. Note that complexity of nonrobust RLS in (3) is also $\mathcal{O}(D_y^3 + D_x^2 D_y)$. It was further argued in the previous remark that OR-RLS does not increase the complexity order. Since the robust schemes in [3], [5], [13], and [16] incur complexity at least as high as that of nonrobust RLS, it follows that they will incur at best the same order of complexity as the proposed OR-RLS.

IV. Convergence Analysis

For the stationary setup, almost sure convergence of OR-RLS to the true $\mathbf{x}_o$ is established in this section under ergodicity conditions on the regressors, nominal noise, and outlier vectors. The following assumptions are needed to this end:

as1) The true parameter vector, denoted as $\mathbf{x}_o$, is time invariant $\forall\, k$.

as2) Three ergodicity conditions hold:

$$\lim_{k\to\infty} \frac{1}{k}\sum_{i=1}^{k} \mathbf{H}_i \mathbf{R}_i^{-1} \mathbf{H}_i^T = \mathbf{R}_\infty > 0, \quad \text{with probability one } (\mathrm{w.p.1}) \tag{12a}$$

$$\lim_{k\to\infty} \frac{1}{k}\sum_{i=1}^{k} \mathbf{H}_i^T \mathbf{R}_k^{-1} \mathbf{v}_i = \mathbf{r}_\infty^{Hv}, \quad (\mathrm{w.p.1}) \tag{12b}$$

$$\lim_{k\to\infty} \frac{1}{k}\sum_{i=1}^{k} \mathbf{H}_i^T \mathbf{R}_k^{-1} \mathbf{o}_i = \mathbf{r}_\infty^{Ho}, \quad (\mathrm{w.p.1}). \tag{12c}$$

If the random vectors $\mathbf{v}_i$, $\mathbf{o}_i$, and $\mathbf{h}_i$ are mixing, which is the case for most stationary processes with vanishing memory in practice, then (12) is satisfied. Since $\mathbf{H}_i$ is uncorrelated with $\mathbf{v}_i$ which is zero-mean, it follows readily that $\mathbf{r}_\infty^{Hv} = \mathbf{0}$.

as3) Regressors $\{\mathbf{h}_i\}$ are zero-mean and uncorrelated with outliers $\{\mathbf{o}_i\}$, which implies that $\mathbf{r}_\infty^{Ho} = \mathbf{0}$.

Since the true $\mathbf{x}_o$ is constant, the forgetting factor is set to $\beta = 1$. The OR-RLS algorithm is also slightly modified to facilitate convergence analysis. Specifically, for k values exceeding a finite value k_1, the estimated outlier vector $\hat{\mathbf{o}}_k$ is deterministically set to zero. This can be achieved by selecting (cf. [6])

$$\begin{aligned} &\lambda_k = \lambda_y,\ k = k_0 + 1, \ldots, k_1, \\ &\lambda_k = \max\left(\lambda_{k-1}, \|\mathbf{R}_k^{-1}(\mathbf{y}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k-1})\|_\infty\right), \quad k > k_1. \end{aligned} \tag{13}$$

Note that the OR-RLS estimate is the minimizer of $\mathcal{J}(\mathbf{x}_k, \hat{\mathbf{o}}_{1:k})$ w.r.t. $\mathbf{x}_k$ with $\mathcal{J}(\cdot)$ defined by (2), and $\hat{\mathbf{o}}_{1:k}$ obtained from (6) with λ_k as in (13). It then follows from as1)–as3) that

$$\lim_{k\to\infty} \frac{\mathcal{J}(\mathbf{x}, \hat{\mathbf{o}}_{1:k})}{k} = \frac{1}{2}\mathbf{x}^T \mathbf{R}_\infty \mathbf{x} - \mathbf{x}^T \left(\mathbf{R}_\infty \mathbf{x}_0 + \mathbf{r}_\infty^{hv} + \mathbf{r}_\infty^{ho}\right) + \lim_{k\to\infty} \frac{1}{k}\sum_{i=1}^{k} \hat{\mathbf{o}}_i^T \mathbf{R}_i^{-1} \mathbf{H}_i \mathbf{x}, \quad (\mathrm{w.p.1}) \tag{14a}$$

$$= \frac{1}{2}\mathbf{x}^T \mathbf{R}_\infty \mathbf{x} - \mathbf{x}^T \mathbf{R}_\infty \mathbf{x}_0, \quad (\mathrm{w.p.1}). \tag{14b}$$

The last term in (14a) becomes zero because $\hat{\mathbf{o}}_i = \mathbf{0}\ \forall\ k > k_1$, and hence the summation assumes a finite value. Setting the gradient of (14b) equal to zero yields $\mathbf{x} = \mathbf{x}_o$, which means that the true $\mathbf{x}_o$ is recovered asymptotically. These arguments establish the following result.

Proposition 1: Under as1)–as3), the modified OR-RLS iterates converge almost surely to the true $\mathbf{x}_o$.

The key observation in the convergence analysis is that even ordinary RLS asymptotically converges to the true $\mathbf{x}_o$. Therefore, one can ensure convergence by making OR-RLS behave like ordinary RLS asymptotically. The intuition behind asymptotic convergence of the ordinary RLS lies with the observation that outliers can be seen as an additional nominal noise contaminating the measurements. Therefore, outlier contaminated measurements can be seen as noncontaminated ones, but with larger nominal noise variance. Furthermore, as3) ensures that no bias appears asymptotically. Hence, when infinitely many measurements are collected, the true $\mathbf{x}_o$ is recovered.

Both RLS and OR-RLS converge almost surely to the true $\mathbf{x}_o$. However, the finite-sample performance of the two algorithms is drastically different. Unlike RLS, OR-RLS maintains satisfactory performance even for a small finite k. Simulations will demonstrate the superior performance of OR-RLS compared to RLS and two robust RLS alternatives.

V. Simulations

A setup with $D_x = 20$ and $D_y = 10$ is tested. In the stationary case, $\mathbf{x}_k = \mathbf{x}_o$, $\forall\ k$ is selected as the all-one vector, with zero-mean, unit-variance, Gaussian noise added. Entries of the regressor matrix $\mathbf{H}_k$ are chosen as zero-mean, white, Gaussian distributed random variables with variance σ_h^2 scaled to achieve an $\mathrm{SNR} = 2D_x\sigma_h^2 = 2$ in normal scale. Per time step k, ambient zero-mean, colored, Gaussian noise $\mathbf{v}_k$ is simulated as the output of a first-order autoregressive (AR) filter with pole at 0.95 to a unit variance white Gaussian input. Vectors $\mathbf{v}_k$ are uncorrelated across k. Measurements are also contaminated by impulsive noise, which generates outliers. When an outlier occurs, which has a 20% chance of happening, the corresponding observation is drawn from a uniform distribution independent of all other variables with variance equal to 20 000. The way outliers are generated differs from the model assumed in (1). This serves to signify the universality of the proposed algorithm as it demonstrates its operability even when the outliers are generated differently than modeled. The root mean-square error RMSE $:= \|\hat{\mathbf{x}}_k - \mathbf{x}_k\|_2$ is used as performance metric. Furthermore, $k_0 = 20$ and k ranges from 1 to 1000. It is assumed that the percentage of outliers is known to all estimators. OR-RLS utilizes this knowledge by using the corresponding criterion from [6] to tune the parameter λ_y.

A. Stationary Scenario

Fig. 1 depicts RMSE versus time for a sample realization. RLS is initialized with the batch LS solution. It can be seen that RLS performs poorly, while OR-RLS performs satisfactorily. Initialization with the batch offline estimator in (4) clearly improves the accuracy of OR-RLS. Different renditions of OR-RLS, which result from the way (6) is solved, are also compared. The best performance is obtained when (6) is solved exactly using SeDuMi. To assess performance of the low-complexity CD-based solver, the RMSE is depicted for 10, 20, 50, and 100 iterations. With 100 CD iterations the performance of OR-RLS comes very close to the SeDuMi based one. Likewise, with 100 AD-MoM iterations, estimation performance comes close to that of SeDuMi as can be verified from Fig. 2. AD-MoM's regularization scale in the augmented Lagrangian is selected as $\kappa = 10$, which being positive ensures convergence due to the convexity of (6). (Convergence rate depends on the κ value; $\kappa = 10$ was observed to maintain a satisfactory convergence rate.) To compare complexities of the various OR-RLS renditions, Table II lists the run-time of each algorithm depicted in Figs. 1 and 2. Caution should be exercised though since run-time comparisons depend on how efficiently each algorithm is programmed in MATLAB. Nonetheless, the table provides an idea of each algorithm's relative complexity. Clearly, AD-MoM and CD based renditions incur lower complexity than the SeDuMi based OR-RLS solver even with 100 AD-MoM or CD iterations. Given the comparable performance of the AD-MoM and CD based solvers

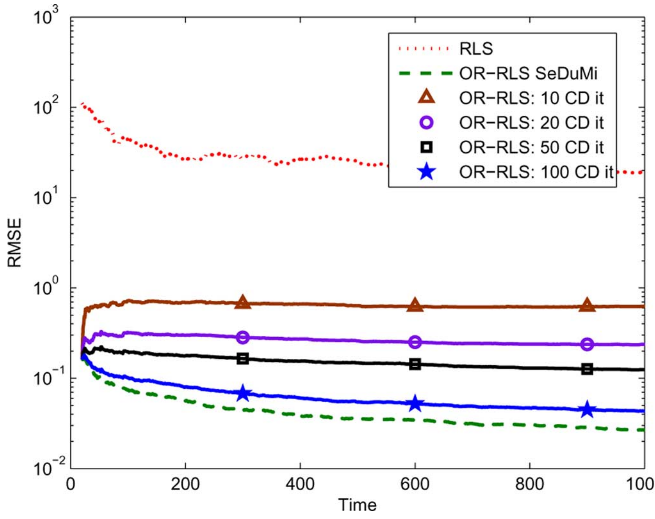

Fig. 1. Learning curves of OR-RLS and its CD-based implementation.

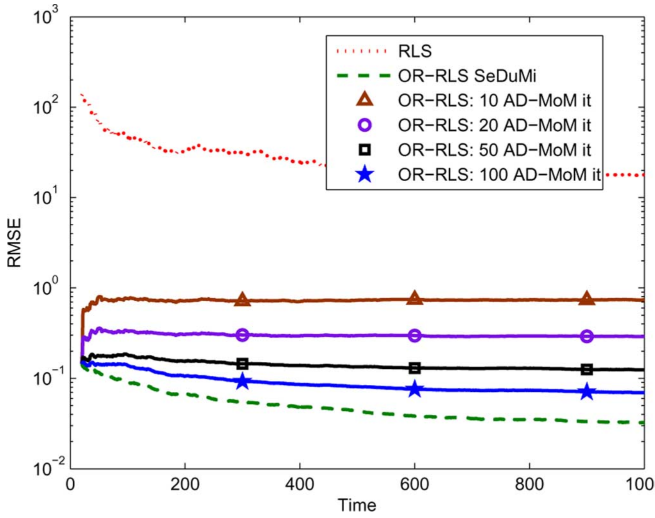

Fig. 2. Learning curves of OR-RLS and its AD-MoM based implementation.

TABLE II
OR-RLS RUN-TIME

Algorithm / Sub-algorithm	Run-time in seconds
RLS	1.0549
OR-RLS w/ SeDuMi	156.8047
OR-RLS w/ AD-MoM 10 it	35.3822
OR-RLS w/ AD-MoM 20 it	35.4259
OR-RLS w/ AD-MoM 50 it	35.5421
OR-RLS w/ AD-MoM 100 it	35.9595
Offline Initialization	33.8649
OR-RLS w/ CD 10 it	37.7787
OR-RLS w/ CD 20 it	40.8145
OR-RLS w/ CD 50 it	48.0724
OR-RLS w/ CD 100 it	58.8621

to that of SeDuMi, their use is well motivated. Note also that the bulk of complexity for AD-MoM and CD based OR-RLS solvers comes from the offline initialization stage which entails multiple SeDuMi runs. One SeDuMi run will be needed for each choice of the sparsity-promoting coefficient λ. Focusing on OR-RLS with 100 AD-MoM iterations for example, the run-time excluding the offline initialization will take about 2 s. This comes close to that of nonrobust RLS, which is about 1 s.

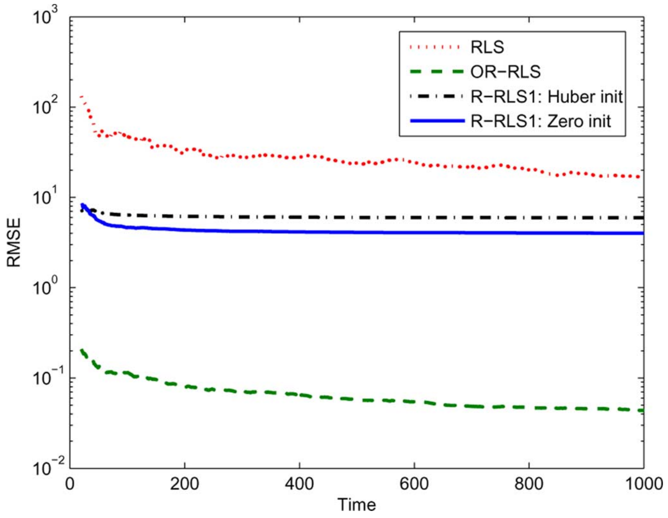

Fig. 3. Comparison between R-RLS1 in [13] and OR-RLS in a stationary setup.

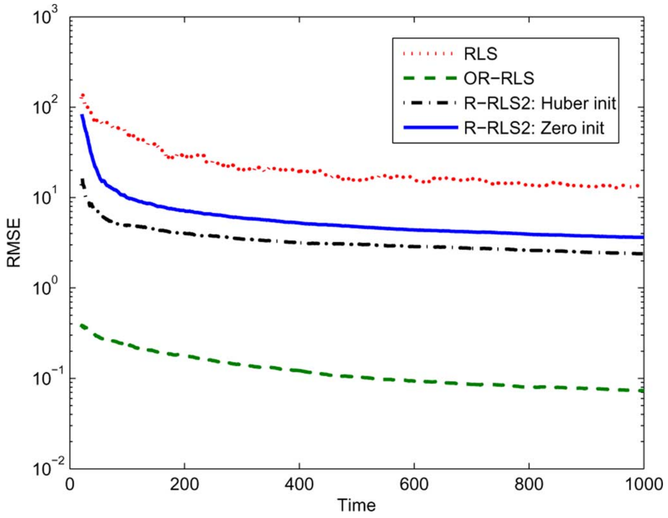

Fig. 4. Comparison between R-RLS2 in [3] and OR-RLS in a stationary setup.

OR-RLS with 100 CD iterations is compared against the robust RLS algorithm of [13], dubbed R-RLS1, and that of [3], dubbed R-RLS2, in Figs. 3 and 4, respectively. Both of these approaches are proposed for white ambient noise, and do not take noise correlations into account. While one can prewhiten the data by multiplying with $\mathbf{R}_k^{-1}$ and then apply R-RLS1 or R-RLS2, this spreads the outliers to all noncontaminated measurements and the resulting performance is as bad as that of RLS. Simulations suggested that the preferred option is to ignore noise correlations which prevents outliers from spreading. This case has been considered in the simulations. To initialize R-RLS1 and R-RLS2 from $k = 1$ up to $k = k_0 = 20$, two different approaches are tested: i) solving the batch Huber cost via iteratively re-weighted LS (IRLS) and ii) starting from the all-zero initialization, and then running R-RLS1 or R-RLS2. The performance of R-RLS1 is depicted in Fig. 3. While better than RLS, R-RLS1 is outperformed by OR-RLS because R-RLS1 does not account for noise correlations. The two initializations perform similarly. Fig. 4 depicts the performance of R-RLS2 versus OR-RLS. Huber initialization provides a considerable improvement in this case, but still OR-RLS exhibits a noticeable improvement over R-RLS2.

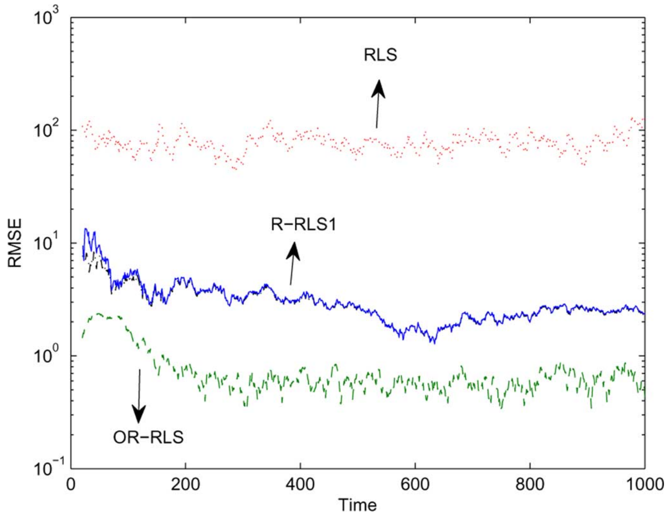

Fig. 5. Comparison between R-RLS1 in [13] and OR-RLS in a nonstationary setting.

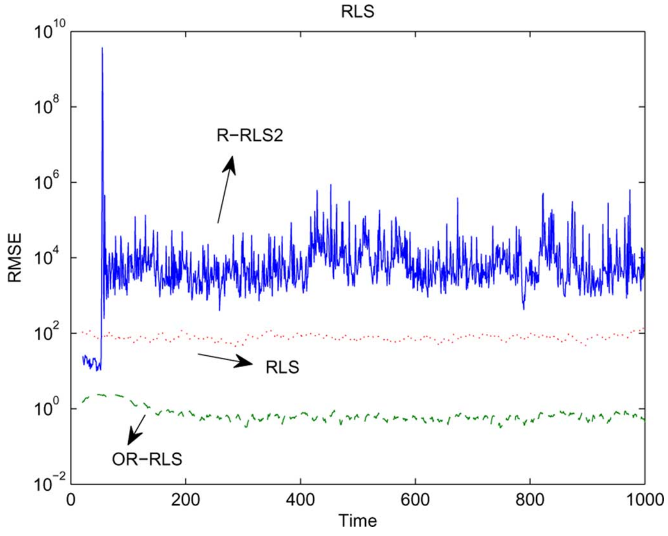

Fig. 6. Comparison between R-RLS2 in [3] and OR-RLS in a nonstationary setting.

B. Non-Stationary Scenario

For the nonstationary case, all parameters including $\mathbf{x}_1$ are selected similar to the stationary case. To obtain $\mathbf{x}_2$, each entry of $\mathbf{x}_1$ is passed through a first-order AR filter with Gaussian noise input. Specifically, letting $x_{k,d}$ denote the dth entry of $\mathbf{x}_k$ the corresponding entry of $\mathbf{x}_{k+1}$ is generated according to $x_{k+1,d} = 0.95x_{k,d} + 0.05e_{k,d}$, where $e_{k,d}$ is zero-mean, unit-variance, Gaussian noise assumed independent of all other noise terms. This procedure is repeated for $k > 2$ to obtain a slow-varying $\mathbf{x}_k$. Figs. 5 and 6 depict the RMSE of OR-RLS with $\beta = 0.95$ versus RLS, R-RLS1 of [13], and R-RLS2 of [3]. It can be seen that OR-RLS markedly outperforms all three alternatives.

VI. Conclusions and Current Research

A robust RLS algorithm was developed capable of simultaneously handling measurement outliers and accounting for possibly correlated nominal noise. To achieve these desirable properties, outliers were treated as nuisance variables and were estimated jointly with the parameter vector of interest. Identifiability was ensured by regularizing the LS cost with the ℓ_1-norm of the outlier variables. The resulting optimization problem, dubbed as offline benchmark, was too complex to be solved per time step. For low-complexity implementation, a suboptimal online algorithm was developed which involved only simple closed-form updates per time step. Initialization with the offline benchmark followed by online updates was advocated as the OR-RLS algorithm of choice in practice. Numerical tests demonstrated improved performance of OR-RLS compared to RLS and state-of-the-art robust RLS renditions.

Current research focuses on robust online estimation of dynamical processes adhering to a Gauss–Markov model, which in addition to slow parameter changes that can be followed by OR-RLS, enables tracking of fast-varying parameters too. Preliminary results in this direction can be found in [6].

Acknowledgment

The authors would like to thank Dr. D. Angelosante of ABB, Switzerland, for helpful discussions and suggestions on the early stages of this work.

References

[1] D. Angelosante, J.-A. Bazerque, and G. B. Giannakis, "Online adaptive estimation of sparse signals: Where RLS meets the 11-norm," *IEEE Trans. Signal Process.*, vol. 58, no. 7, pp. 3436–3447, Jul. 2010.

[2] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*, 2nd ed. Belmont, MA: Athena Scientific, 1999.

[3] M. Z. A. Bhotto and A. Antoniou, "Robust recursive least-squares adaptive-filtering algorithms for impulsive-noise environments," *IEEE Signal Process. Lett.*, vol. 18, no. 3, pp. 185–188, Mar. 2011.

[4] E. J. Candes and T. Tao, "Decoding by linear programming," *IEEE Trans. Inf. Theory*, vol. 51, no. 12, pp. 4203–4215, Dec. 2005.

[5] S. C. Chan and Y. X. Zou, "A recursive least M-estimate algorithm for robust adaptive filtering in impulsive noise: Fast algorithm and convergence performance analysis," *IEEE Trans. Signal Process.*, vol. 52, no. 4, pp. 975–991, Apr. 2004.

[6] S. Farahmand, D. Angelosante, and G. B. Giannakis, "Doubly Robust Kalman smoothing by exploiting outlier sparsity," presented at the Asilomar Conf. Signals, Syst., Comput., Pacific Grove, CA, Nov. 2010.

[7] J.-J. Fuchs, "An inverse problem approach to robust regression," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, Phoenix, AZ, 1999, pp. 1809–1812.

[8] G. B. Giannakis, G. Mateos, S. Farahmand, V. Kekatos, and H. Zhu, "USPACOR: Universal sparsity-controlling outlier rejection," presented at the Int. Conf. Acoust., Speech, Signal Process., Prague, Czech Republic, May 2011.

[9] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, 2nd ed. New York: Springer, 2009.

[10] P. J. Huber and E. M. Ronchetti, *Robust Statistics*. New York: Wiley, 2009.

[11] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*. Englewood Cliffs, NJ: Prentice-Hall, 1993.

[12] P. Petrus, "Robust Huber adaptive filter," *IEEE Trans. Signal Process.*, vol. 47, no. 4, pp. 1129–1133, Apr. 1999.

[13] L. R. Vega, H. Rey, J. Benesty, and S. Tressens, "A fast robust recursive least-squares algorithm," *IEEE Trans. Signal Process.*, vol. 57, no. 3, pp. 1209–1216, Mar. 2009.

[14] SeDuMi [Online]. Available: http://sedumi.ie.lehigh.edu/

[15] P. Tseng, "Convergence of a block coordinate descent method for nondifferentiable minimization," *J. Optim. Theory Appl.*, vol. 109, no. 3, pp. 475–494, Jun. 2001.

[16] Y. Zou, S. C. Chan, and T. S. Ng, "A recursive least M-estimate (RLM) adaptive filter for robust filtering in impulsive noise," *IEEE Signal Process. Lett.*, vol. 7, no. 11, pp. 324–326, Nov. 2000.