西南交通大學
Southwest Jiaotong University

Robust Diffusion Recursive Least Squares Estimation with Side Information for Networked Agents

Yi Yu, Haiquan Zhao, Rodrigo C. de Lamare, and Yuriy Zakharov

April 20, 2018

School of Electrical Engineering, Southwest Jiaotong University, China

Outline

- Introduction
- Problem Formulation
- Proposed Robust dRLS algorithm
- Simulation results
- Conclusions

Introduction

➢ Distributed estimation

- Distributed adaptive algorithms are of great attention for estimating parameters of interest in wireless sensor networks.
- Such techniques is to perform the parameter estimation from data collected from nodes (or agents) in-network.
- The basic idea is that each node performs adaptive estimation in cooperation with its neighboring nodes.

  [R1] A.H. Sayed, "Adaptation, learning, and optimization over networks," Foundations and Trends in Machine Learning, vol. 7, no. 4-5, pp. 311–801, 2014.

- Distributed adaptive algorithms have been applied to many problems, e.g., frequency estimation in power grid, and spectrum estimation.

  [R2] S. Kanna, D.H. Dini, Y. Xia, S.Y. Hui, and D.P. Mandic, "Distributed widely linear kalman filtering for frequency estimation in power networks," IEEE Transactions on Signal and Information Processing over Networks, vol. 1, no. 1, pp. 45–57, 2015.

  [R3] T.G. Miller, S. Xu, R.C. de Lamare, and H.V. Poor, "Distributed spectrum estimation based on alternating mixed discrete-continuous adaptation," IEEE Signal Processing Letters, vol. 23, no. 4, pp. 551–555, 2016.

Introduction

➢ Distributed algorithms

- According to the cooperation way of interconnected nodes, existing algorithms can be categorized as the incremental, consensus, and diffusion types.
- The diffusion protocol is the most popular, because it does not require a Hamiltonian cycle path as does the incremental type; it is stable and has a better estimation performance than the consensus type.

  [R4] S.Y. Tu and A.H. Sayed, “Diffusion strategies outperform consensus strategies for distributed estimation over adaptive networks,” IEEE Transactions on Signal Processing, vol. 60, no. 12, pp. 6217–6234, 2012.
- Several diffusion-based distributed algorithms have been proposed, e.g., the diffusion least mean square (dLMS) algorithm, diffusion recursive least squares (dRLS) algorithm, and their modifications.

Introduction

➢ Existing robust ways against impulsive noises

- In practice, measurements at the network nodes can be corrupted by impulsive noise. Impulsive noise has the property that its occurence probability is small and magnitude is typically much higher than the nominal measurement.

  [R5] K.L. Blackard, T.S. Rappaport, and C.W. Bostian, "Measurements and models of radio frequency impulsive noise for indoor wireless communications," IEEE Journal on selected areas in communications, vol. 11, no. 7, pp. 991–1001, 1993.

- Impulsive noise deteriorates significantly the performance of many algorithms in the single-agent case.
- In addition, for distributed algorithms in the multi-agent case, the adverse effect of impulsive noise at one node can also propagate over the entire network due to the exchange of information among nodes.

Introduction

- Aiming to impulsive noise scenarios, many robust distributed algorithms have been proposed.
- Some algorithms, e.g., the diffusion sign error LMS (dSE-LMS), are based on using the instantaneous gradient-descent method to minimize an individual robust criterion.

[R6] J. Ni, J. Chen, and X. Chen, "Diffusion sign-error LMS algorithm: Formulation and stochastic behavior analysis," Signal Processing, vol. 128, pp. 142–149, 2016.

- A robust variable weighting coefficients dLMS (RVWC-dLMS) algorithm was developed, which only considers the data and intermediate estimates from nodes not affected by impulsive.

[R7] D.C. Ahn, J.W. Lee, S.J. Shin, and W.J. Song, "A new robust variable weighting coefficients diffusion LMS algorithm," Signal Processing, vol. 131, pp. 300–306, 2017.

- However, these robust algorithms have slow convergence, especially for colored input signals at nodes.

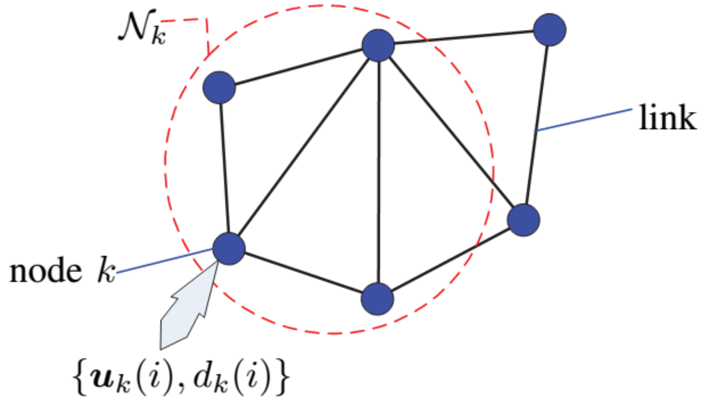
Introduction

➢ Our contributions

- We present a robust dRLS (R-dRLS) algorithm, which is robust against impulsive noise and provides good decorrelating property for colored input signals.
- The R-dRLS algorithm minimizes a local exponentially weighted least squares (LS) cost function subject to a time-dependent constraint on the squared norm of the intermediate estimate at each node.
- In order to equip the R-dRLS algorithm with the ability to withstand sudden changes in the environment, we also propose a diffusion-based distributed nonstationary control (DNC) method.

Problem Formulation

- Consider a network that has N nodes distributed over some region in space.



Diffusion network

where,

k - node index,

i - time instant,

$\mathcal{N}_k$ - neighborhood of node k, i.e., a set of all nodes connected to node k including itself,

n_k - cardinality of $\mathcal{N}_k$

Problem Formulation

- At every time instant $i \geq 0$, node *k* has an input vector $\boldsymbol{u}_{k,i}$ with *M*-dimension and a desired output $d_k(i)$, related as:

$$d_k(i) = \boldsymbol{u}_{k,i}^T \boldsymbol{w}^o + v_k(i) \quad (1)$$

where,

$v_k(i)$ - additive noise,

$\boldsymbol{w}^o$ - parameter vector of size $M \times 1$

- The task is to estimate $\boldsymbol{w}^o$ using the available data collected at nodes, i.e., $\{\boldsymbol{u}_{k,i}, d_k(i)\}_{k=1}^N$.

Problem Formulation

- For this purpose, the global LS-based estimation problem is described as [R8]:

$$\boldsymbol{w}_i = \arg\min_{\boldsymbol{w}} \left\{ \lambda^{i+1}\delta\|\boldsymbol{w}\|_2^2 + \sum_{j=0}^{i} \lambda^{i-j} \sum_{k=1}^{N} \left(d_k(j) - \boldsymbol{u}_{k,i}^T \boldsymbol{w} \right)^2 \right\}, \quad (2)$$

where,

$\|\cdot\|_2$ - the l_2-norm of a vector,

$\delta > 0$ - the regularization constant,

λ - the forgetting factor.

- The dRLS algorithm solves (2) in a distributed manner.

[R8] F.S. Cattivelli, C.G. Lopes, and A.H. Sayed, “Diffusion recursive least-squares for distributed estimation over adaptive networks,” IEEE Transactions on Signal Processing, vol. 56, no. 5, pp. 1865–1877, 2008.

Problem Formulation

- In practice, $v_k(i)$ may contain impulsive noise, severely corrupting the desired output $d_k(i)$.

- With such noise processes, the algorithms obtained from (2), e.g., the dRLS algorithm, would fail to work.

Proposed R-dRLS algorithm

➢ Derivation of algorithm

- We focus here on the adapt-then-combine (ATC) implementation of the diffusion strategy, which has been shown to outperform the combine-then-adapt (CTA) implementation.

- In fact, the CTA version is obtained by reversing the adaptation step and combination step in the ATC version.

➢ Step 1: we start with the adaptation step.

Proposed R-dRLS algorithm

Every node k, at time instant i, finds an intermediate estimate $\boldsymbol{\psi}_{k,i}$ of $\boldsymbol{w}^o$ by minimizing the individual local cost function:

$$J_k(\boldsymbol{\psi}_{k,i}) = \|\boldsymbol{\psi}_{k,i} - \boldsymbol{w}_{k,i-1}\|^2_{\boldsymbol{Q}_{k,i}} + [d_k(i) - \boldsymbol{u}^T_{k,i}\boldsymbol{\psi}_{k,i}]^2, \tag{3}$$

with $\boldsymbol{Q}_{k,i} = \boldsymbol{R}_{k,i} - \boldsymbol{u}_{k,i}\boldsymbol{u}^T_{k,i}$, where

$$\begin{aligned} \boldsymbol{R}_{k,i} &\triangleq \lambda^{i+1}\delta\boldsymbol{I} + \sum_{j=0}^{i} \lambda^{i-j}\boldsymbol{u}_{k,j}\boldsymbol{u}^T_{k,j} \\ &= \lambda\boldsymbol{R}_{k,i-1} + \boldsymbol{u}_{k,i}\boldsymbol{u}^T_{k,i} \end{aligned} \tag{4}$$

is the time-averaged correlation matrix for the regression vector at node k, $\boldsymbol{w}_{k,i-1}$ is an estimate of $\boldsymbol{w}^o$ at node k at time instant $i-1$, and $\boldsymbol{I}$ is the identity matrix. Notice that the form $\|\boldsymbol{x}\|^2_{\boldsymbol{Q}} \triangleq \boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x}$ in (3) defines the Riemmanian distance between vectors $\boldsymbol{\psi}_{k,i}$ and $\boldsymbol{w}_{k,i-1}$.

Proposed R-dRLS algorithm

- Setting the derivative of $J_k(\boldsymbol{\psi}_{k,i})$ with respect to $\boldsymbol{\psi}_{k,i}$ to zero, we obtain

$$\boldsymbol{\psi}_{k,i} = \boldsymbol{w}_{k,i-1} + \boldsymbol{P}_{k,i}\boldsymbol{u}_{k,i}e_k(i), \tag{5}$$

where $e_k(i) = d_k(i) - \boldsymbol{u}_{k,i}^T\boldsymbol{w}_{k,i-1}$ stands for the output error at node k and $\boldsymbol{P}_{k,i} \triangleq \boldsymbol{R}_{k,i}^{-1}$. Using the matrix inversion lemma, we have

$$\boldsymbol{P}_{k,i} = \frac{1}{\lambda}\left(\boldsymbol{P}_{k,i-1} - \frac{\boldsymbol{P}_{k,i-1}\boldsymbol{u}_{k,i}\boldsymbol{u}_{k,i}^T\boldsymbol{P}_{k,i-1}}{\lambda + \boldsymbol{u}_{k,i}^T\boldsymbol{P}_{k,i-1}\boldsymbol{u}_{k,i}}\right), \tag{6}$$

where $\boldsymbol{P}_{k,i}$ is initialized as $\boldsymbol{P}_{k,0} = \delta^{-1}\boldsymbol{I}$.

- Obviously, the adverse effect of an impulsive noise sample at time instant i will propagate via $e_k(i)$.

Proposed R-dRLS algorithm

- To make the algorithm robust in impulsive noise scenarios, we propose to minimize (3) under the following constraint:

$$\boxed{\|\boldsymbol{\psi}_{k,i} - \boldsymbol{w}_{k,i-1}\|_2^2 \leq \xi_k(i-1),} \tag{7}$$

where $\xi_k(i-1)$ is a positive bound. This constraint is employed to enforce the squared norm of the update of the intermediate estimate not to exceed the amount $\xi_k(i-1)$ regardless of the type of noise (possibly, impulsive noise), thereby guaranteeing the robustness of the algorithm.

- If (5) satisfies (7), i.e.,

$$\|\boldsymbol{g}_{k,i}\|_2 |e_k(i)| \leq \sqrt{\xi_k(i-1)}, \tag{8}$$

where $\boldsymbol{g}_{k,i} \triangleq \boldsymbol{P}_{k,i}\boldsymbol{u}_{k,i}$ represents the Kalman gain vector, then (5) is a solution of the above constrained minimization problem.

Proposed R-dRLS algorithm

- If (8) is not satisfied (usually in the case of appearance of impulsive noise samples), we propose to the following normalized update to replace (5),

$$\boldsymbol{\psi}_{k,i} = \boldsymbol{w}_{k,i-1} + \sqrt{\xi_k(i-1)} \frac{\boldsymbol{g}_{k,i}}{\|\boldsymbol{g}_{k,i}\|_2} \mathrm{sign}(e_k(i)), \tag{9}$$

where sign(·) is the sign function. Obviously, (9) satisfies the equal sign in the constraint (7).

- Combining (5), (8) and (9), we obtain the adaptation step for each node *k* as:

$$\boldsymbol{\psi}_{k,i} = \boldsymbol{w}_{k,i-1} + \min\left[\frac{\sqrt{\xi_k(i-1)}}{\|\boldsymbol{g}_{k,i}\|_2 |e_k(i)|}, 1\right] \boldsymbol{g}_{k,i} e_k(i). \tag{10}$$

Proposed R-dRLS algorithm

- Step 2: the intermediate estimates $\psi_{m,i}$ from the neighborhood $m \in \mathcal{N}_k$ of node *k* linearly weighted, yielding a more reliable estimate:

$$\boldsymbol{w}_{k,i} = \sum_{m \in \mathcal{N}_k} c_{m,k} \boldsymbol{\psi}_{m,i}, \tag{11}$$

where the combination coefficients $\{c_{m,k}\}$ are non-negative, and satisfy:

$$\sum_{m \in \mathcal{N}_k} c_{m,k} = 1, \text{ and } c_{m,k} = 0 \text{ if } m \notin \mathcal{N}_k. \tag{12}$$

- $c_{m,k}$ denotes the weight assigned by node *k* to its neighbor intermediate $\boldsymbol{\psi}_{m,i}$. In this paper, $\{c_{m,k}\}$ are determined by a static rule.

Proposed R-dRLS algorithm

- Step 3: to further improve the performance, we propose to recursively adjust $\xi_k(i)$ as:

$$\begin{aligned} \zeta_k(i) =& \beta\xi_k(i-1) + (1-\beta)\left\|\boldsymbol{\psi}_{k,i} - \boldsymbol{w}_{k,i-1}\right\|_2^2 \\ =& \beta\xi_k(i-1) + (1-\beta)\min[\|\boldsymbol{g}_{k,i}\|_2^2 e_k^2(i), \xi_k(i-1)], \\ \xi_k(i) =& \sum_{m\in\mathcal{N}_k} c_{m,k}\zeta_m(i), \end{aligned} \tag{13}$$

where, β ($0<\beta<1$) is a forgetting factor.

In (13), $\xi_k(i)$ is initialized as $\xi_k(0) = E_c\sigma_{d,k}^2/(M\sigma_{u,k}^2)$, E_c is a positive integer, and $\sigma_{d,k}^2$ and $\sigma_{u,k}^2$ are powers of signal $d_k(i)$ and $\boldsymbol{u}_{k,i}$, respectively.

Proposed R-dRLS algorithm

Performance explanation

- (10) shows that the operation mode of the proposed algorithm is twofold.
- At the early iterations, compared with $\|\boldsymbol{g}_{k,i}\|_2^2 e_k^2(i)$, the value of $\xi_k(i)$ can be high so that the algorithm will behave as the dRLS algorithm.
- Whenever an impulsive noise sample appears, due to its significant magnitude, the algorithm will work as a dRLS update multiplied by a very small 'step size' scaling factor given by $\sqrt{\xi_k(i-1)}/(\|\boldsymbol{g}_{k,i}\|_2|e_k(i)|)$, thus avoiding the negative influence of impulsive noise on the estimation.
- $\xi_k(i)$ computed by (13) over the iterations is decreasing over the iterations, thus further improving the algorithm robustness against impulsive noise.

Proposed R-dRLS algorithm

➢ DNC method

To improve the tracking capability of the algorithm for a sudden change of the parameter vector, inspired by the single-agent scenario [R9], we propose the DNC method.

Firstly, a variable $\Delta_k(i)$ at node k is computed once for every V_t iterations, to judge whether the unknown vector has a change or not. In this step, $\boldsymbol{a}_{k,i}^T = \mathcal{O}\left(\left[\frac{e_k^2(i)}{\|\boldsymbol{u}_{k,i}\|_2^2}, \frac{e_k^2(i-1)}{\|\boldsymbol{u}_{k,i-1}\|_2^2}, ..., \frac{e_k^2(i-V_t+1)}{\|\boldsymbol{u}_{k,i-V_t+1}\|_2^2}\right]\right)$ with $\mathcal{O}(\cdot)$ denoting the ascending arrangement for its arguments, and $\boldsymbol{e} = [1, ..., 1, 0, ..., 0]^T$ is a vector whose first $V_t - V_d$ elements set to one, where V_d is a positive integer with $V_d < V_t$. Thus, the product $\boldsymbol{a}_{k,i}^T\boldsymbol{e}$ can reduce the effect of outliers (e.g., impulsive noise samples) when computing $\Delta_k(i)$. Typically, for both V_t and V_d, good choices are $V_t = \varrho M$ with $\varrho = 1 \sim 3$ and $V_d = 0.75V_t$. Note that, for larger occurence probability of impulsive noise, the value of $V_t - V_d$ should be decreased to discard the impulsive noise samples.

Proposed R-dRLS algorithm

Secondly, if $\Delta_k(i) > t_{\text{th}}$, where t_{th} is a predefined threshold, meaning a change of $\boldsymbol{w}^o$ has occured, then we need to reset $\xi_k(i)$ to its initial value $\xi_k(0)$. More importantly, $\boldsymbol{P}_{k,i}$ is also re-initialized with $\boldsymbol{P}_{k,0}$. It is worth noting that since the parameters γ, N_w, ϱ, and t_{th} are not affected by each other, their choices are simplified.

- The proposed R-dRLS algorithm with the DNC method is summarized in Table 1.

[R9] L.R. Vega, H. Rey, J. Benesty, and S. Tressens, "A new robust variable step-size NLMS algorithm," IEEE Transactions on Signal Processing, vol. 56, no. 5, pp. 1878–1893, 2008.

Proposed R-dRLS algorithm

Table 1. Proposed R-dRLS Algorithm with the DNC Method.

Parameters: $0 < \beta \lesssim 1$, λ, δ and E_c (R-dRLS); ϱ and t_{th} (DNC)

Initialization: $\boldsymbol{w}_{k,0} = \boldsymbol{0}$, $\boldsymbol{P}_{k,0} = \delta^{-1}\boldsymbol{I}$ and $\xi_k(0) = E_c \frac{\sigma^2_{d,k}}{M\sigma^2_{u,k}}$ (R-dRLS)

$\Theta_{\text{old},k} = \Theta_{\text{new},k} = 0$, $V_t = \varrho M$, and $V_d = 0.75V_t$ (DNC)

R-dRLS algorithm:

$e_k(i) = d_k(i) - \boldsymbol{u}^T_{k,i}\boldsymbol{w}_{k,i-1}$

$\boldsymbol{P}_{k,i} = \frac{1}{\lambda}\left(\boldsymbol{P}_{k,i-1} - \frac{\boldsymbol{P}_{k,i-1}\boldsymbol{u}_{k,i}\boldsymbol{u}^T_{k,i}\boldsymbol{P}_{k,i-1}}{\lambda+\boldsymbol{u}^T_{k,i}\boldsymbol{P}_{k,i-1}\boldsymbol{u}_{k,i}}\right)$

$\boldsymbol{g}_{k,i} = \boldsymbol{P}_{k,i}\boldsymbol{u}_{k,i}$

$\boldsymbol{\psi}_{k,i} = \boldsymbol{w}_{k,i-1} + \min\left[\frac{\sqrt{\xi_k(i-1)}}{\|\boldsymbol{g}_{k,i}\|_2|e_k(i)|},\ 1\right]\boldsymbol{g}_{k,i}e_k(i)$

$\boldsymbol{w}_{k,i} = \sum_{m\in\mathcal{N}_k} c_{m,k}\boldsymbol{\psi}_{m,i}$

DNC method:

Step 1: to compute $\Delta_k(i)$

if $i = nV_t, n = 0, 1, 2, \ldots$

$\boldsymbol{a}^T_{k,i} = \mathcal{O}\left(\left[\frac{e^2_k(i)}{\|\boldsymbol{u}_{k,i}\|^2_2}, \frac{e^2_k(i-1)}{\|\boldsymbol{u}_{k,i-1}\|^2_2}, \ldots, \frac{e^2_k(i-V_t+1)}{\|\boldsymbol{u}_{k,i-V_t+1}\|^2_2}\right]\right)$

$\Theta_{\text{new},k} = \sum_{m\in\mathcal{N}_k} c_{m,k}\frac{\boldsymbol{a}^T_{m,i}\boldsymbol{e}}{V_t - V_d}$

$\Delta_k(i) = \frac{\Theta_{\text{new},k} - \Theta_{\text{old},k}}{\xi_k(i-1)}$

end

Step 2: to reset $\xi_k(i)$

if $\Delta_k(i) > t_{\text{th}}$

$\zeta_k(i) = \xi_k(0)$, $\boldsymbol{P}_{k,i} = \boldsymbol{P}_{k,0}$

elseif $\Theta_{\text{new},k} > \Theta_{\text{old},k}$

$\zeta_k(i) = \xi_k(i-1) + (\Theta_{\text{new},k} - \Theta_{\text{old},k})$

else

$\zeta_k(i) = \beta\xi_k(i-1) + (1-\beta)\min\left[\|\boldsymbol{g}_{k,i}\|^2_2 e^2_k(i),\ \xi_k(i-1)\right]$

end

$\xi_k(i) = \sum_{m\in\mathcal{N}_k} c_{m,k}\zeta_m(i)$

$\Theta_{\text{old},k} = \Theta_{\text{new},k}$

Simulation results

- A diffusion network with N=20 nodes is considered.
- The parameter vector $\boldsymbol{w}^o$ of the length M=16 is generated randomly from a zero-mean uniform distribution, with a unit norm.
- To evaluate the tracking capability, $\boldsymbol{w}^o$ changes to $-\boldsymbol{w}^o$ at the middle of iterations.
- The input vector has a shifted structure, i.e., $\boldsymbol{u}_{k,i} = [u_k(i), u_k(i-1), \ldots, u_k(i-M+1)]^T$, where $u_k(i)$ is colored and generated by a second-order autoregressive system $u_k(i) = 1.6u_k(i-1) - 0.81u_k(i-2) + \epsilon_k(i)$, with $\epsilon_k(i)$ being a zero-mean white Gaussian process with variance $\sigma^2_{\epsilon,k}$.
- The averaged network mean square deviation is used for assessing the algorithm performance, i.e., $\mathrm{MSD}_{\mathrm{net}}(i) = \frac{1}{N}\sum_{k=1}^{N} E\{\|\boldsymbol{w}^o - \boldsymbol{w}_{k,i}\|_2^2\}$.
- All results are the average over 200 independent trials.

Simulation results

➤ Bernoulli-Gaussian (BG) process

- The additive noise $v_k(i)$ includes the background noise $\theta_k(i)$ plus the impulsive noise $\eta_k(i)$, where $\theta_k(i)$ is zero-mean white Gaussian noise with variance $\sigma^2_{\theta,k}$.
- Fig. 1 gives the values of $\sigma^2_{\epsilon,k}$ and $\sigma^2_{\theta,k}$ at all nodes.

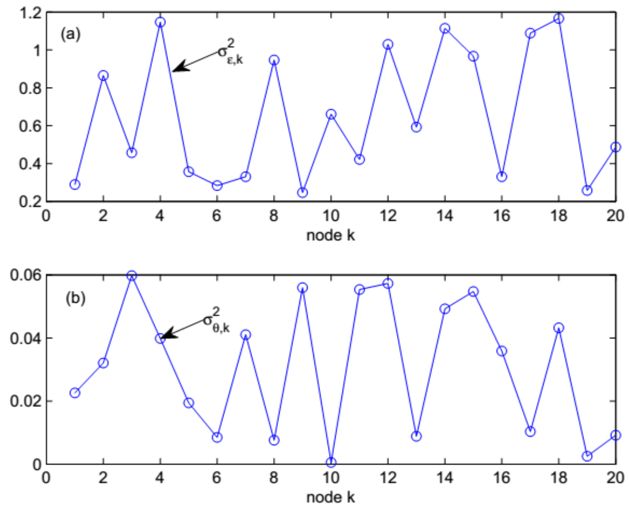


Fig. 1. Profiles of $\sigma^2_{\epsilon,k}$ and $\sigma^2_{\theta,k}$.

The impulsive noise $\eta_k(i)$ is described by the BG process, $\eta_k(i) = b_k(i) \cdot g_k(i)$, where $b_k(i)$ is a Bernoulli process with probability distribution $P[b_k(i) = 1] = p_{r,k}$ and $P[b_k(i) = 0] = 1 - p_{r,k}$, and $g_k(i)$ is a zero-mean white Gaussian process with variance $\sigma^2_{g,k}$. Here, we set $p_{r,k}$ as a random number in the range of $[0.001, 0.05]$, and $\sigma^2_{g,k} = 1000\sigma^2_{y,k}$, where $\sigma^2_{y,k}$ denotes the power of $y_k(i) = \boldsymbol{u}^T_{k,i}\boldsymbol{w}^o$.

Simulation results

- The R-dRLS (no cooperation) algorithm performs an independent estimation at each node. For RLS-type algorithms, we choose λ=0.995 and δ=0.01.
- As expected, the dRLS algorithm has a poor performance in the presence of impulsive noise.
- Both the dSE-LMS and RVWC-dLMS algorithms are significantly less sensitive to impulsive noise, but their convergence is slow.
- Apart from the robustness against impulsive noise, the proposed R-dRLS algorithm has also a fast convergence.
- The proposed DNC method can retain the good tracking capability of the R-dRLS algorithm, only with a slight degradation in steady-state performance.

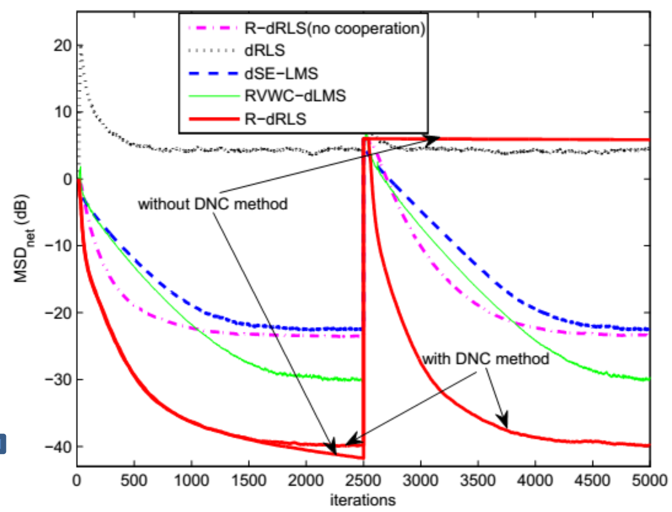


Fig. 2. Averaged network MSD performance of the algorithms in impulsive noise with BG process. Parameter setting of the algorithms (with notations from references) is as follows: μ_k=0.015 (dSE-LMS); β=0.98 and E_c=1 (R-dRLS); ϱ=3 and t_{th}=25 (DNC). For the RVWC-dLMS, the Metropolis rule is also used for the combination coefficients in the adaptation step; its other parameters are L=16, α=2.58, λ=0.98 and μ_k=0.03.

Simulation results

➢ α-Stable process

- The impulsive noise is modeled by the α-stable process with a characteristic function $\varphi(t) = \exp(-\gamma|t|^{\alpha})$ where the characteristic exponent $\alpha \in (0, 2]$ describes the impulsiveness of the noise (smaller α leads to more impulsive noise samples) and $\gamma > 0$ represents the dispersion level of the noise.
- In this example, thus we set $\alpha = 1.15$ and $\gamma = 1/15$.
- Fig. 4 shows the node-wise steady-state MSD of the robust algorithms (i.e., excluding the dRLS) against impulsive noise, by averaging over 500 instantaneous MSD values in the steady-state.
- As can be seen from Figs. 3 and 4, the proposed R-dRLS algorithm with DNC outperforms the known robust algorithms.

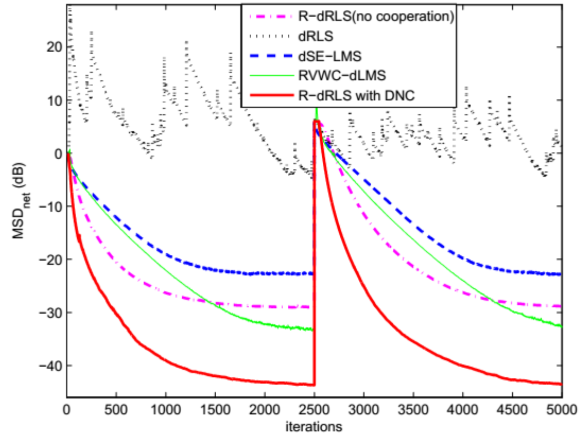


Fig. 3. Averaged network MSD performance of the algorithms in α-Stable noise. Parameter setting is the same as in Fig. 2.

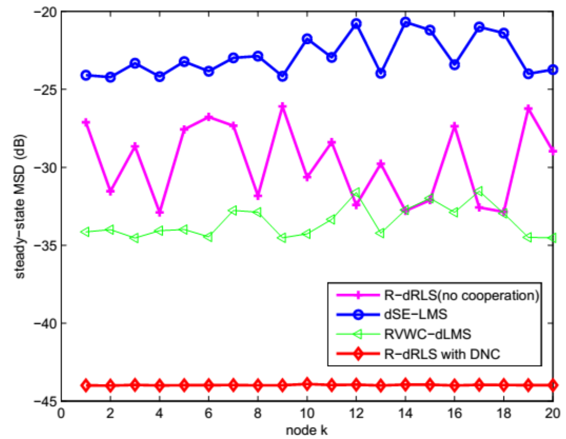


Fig. 4. Node-wise steady-state MSD of the algorithms in α-Stable noise.

Conclusions

- In this paper, the R-dRLS algorithm has been proposed, based on the minimization of an individual RLS cost function with a time-dependent constraint on the squared norm of the intermediate estimate update.
- The constraint is dynamically adjusted based on the diffusion strategy with the help of side information.
- The novel algorithm not only is robust against impulsive noise, but also has fast convergence.
- Furthermore, to track the change of parameters of interest, a detection method (DNC method) is proposed for re-initializing the constraint.
- Simulation results have verified that the proposed algorithm performs better than known algorithms in impulsive noise scenarios.

Thanks

Please feel free contact me at *hqzhao_swjtu@126.com*, if you have any further questions.