Variable-Regularized Low-Complexity RLS Adaptive Algorithms for Bilinear Forms

Ionuț-Dorinel Fîciu*, *Camelia Elisei-Iliescu, Cristian-Lucian Stanciu and Constantin Paleologu,*

Departament of Telecommunications
University Politehnica of Bucharest, Romania
ionut_dorinel.ficiu@upb.ro*,
camelia.elisei@romatsa.ro,
{cristian, pale}@comm.pub.ro

** Presenter*

Presenter’s Biography

- **Current:**
  - **PhD student**
    - @ Doctoral School of Electronics, Telecommunications & Information Technology, University Politehnica of Bucharest since October 2020
    - **Thesis subject:** Efficient algorithms for acoustic applications
    - **Coordinator:** Prof. Constantin Paleologu
- **Past:**
  - **Master's degree**
    - @ Advanced Digital Imaging Techniques (TAID), University Politehnica of Bucharest (2018 - 2020)
    - **Dissertation thesis:** Deep neural networks for environmental sounds classification
    - **Coordinators:** Assoc. Prof. Cristian-Lucian Stanciu, Assoc. Prof. Cristian Anghel
  - **Batchelor's degree** (Valedictorian)
    - @ Telecommunications Technologies and Systems (TST), University Politehnica of Bucharest (2014 - 2018)
    - **Diploma thesis:** Convolutional Neural Networks for Object Segmentation and Tracking in Video Sequences
    - **Coordinators:** Prof. Mihai Ciuc, PhD. Cosmin Toca

Outline

- Introduction
- RLS Algorithm for Bilinear Forms
- RLS-DCD Algorithm for Bilinear Forms
- Regularized RLS Algorithm for Bilinear Forms
- Variable-Regularized RLS Algorithm for Bilinear Forms
- Variable-Regularized RLS-DCD Algorithm for Bilinear Forms
- Simulation Results
- Conclusions

Introduction

- **Recursive least-square (RLS) algorithm → frequently used in system identification problems**

→ the reference (desired) signal:

$$d(n) = \mathbf{r}^T \mathbf{x}(n) + w(n)$$

$\mathbf{r}$ = unknown system (length L)

$\mathbf{x}(n) = [x(n)\ \ x(n-1)\ \ \ldots\ \ x(n-L+1)]^T$

$w(n)$ = system noise

- **In this work → identification of bilinear forms**

[Benesty et al., *IEEE Signal Processing Letters*, May 2017]
[Paleologu et al., *Digital Signal Processing*, April 2018]

→ the reference (desired) signal:

$$d(n) = \mathbf{h}^T \mathbf{X}(n) \mathbf{g} + w(n)$$

$\mathbf{h}, \mathbf{g}$ = unknown systems (length L *and* M)

$\mathbf{X}(n) = [\mathbf{x}_1(n)\ \mathbf{x}_2(n)\ \ \ldots\ \mathbf{x}_M(n)]$

$\mathbf{x}_m(n) = [x_m(n)\ x_m(n-1)\ \ \ldots\ x_m(n-L+1)]^T$

- **Target**

→ Variable-Regularized RLS algorithms for the identification of bilinear forms

Introduction

Model

$$d(n) = \mathbf{h}^T\mathbf{X}(n)\mathbf{g} + w(n)$$

➡ **Bilinear form**
(with respect to the impulse responses)

Examples of applications:

- multi-channel equalization
- nonlinear acoustic echo cancellation

[Gesbert and Duhamel, *IEEE WSSAP*, 1996]
[Huang et al., *IEEE ICASSP*, 2017]
[Stenger and Kellerman, *Signal Processing*, Sept. 2000]

Equivalent model

$$d(n) = \mathbf{f}^T\tilde{\mathbf{x}}(n) + w(n)$$

$\mathbf{f} = \mathbf{g} \otimes \mathbf{h}$ → *Kronecker product*

$$\tilde{\mathbf{x}}(n) = \mathrm{vec}[\mathbf{X}(n)] = \begin{bmatrix} \mathbf{x}_1(n) \\ \mathbf{x}_2(n) \\ \vdots \\ \mathbf{x}_M(n) \end{bmatrix}$$

f → length *ML*
h → length *L*
g → length *M*

RLS Algorithm for Bilinear Forms

The normal equations (LS criterion):

$$\mathbf{R}_{\hat{\mathbf{g}}}(n)\hat{\mathbf{h}}(n) = \mathbf{p}_{\hat{\mathbf{g}}}(n)$$
$$\mathbf{R}_{\hat{\mathbf{h}}}(n)\hat{\mathbf{g}}(n) = \mathbf{p}_{\hat{\mathbf{h}}}(n)$$

, where

$$\mathbf{R}_{\hat{\mathbf{g}}}(n)=\lambda_{\hat{\mathrm{h}}}\mathbf{R}_{\hat{\mathbf{g}}}(n-1)+\tilde{\mathbf{x}}_{\hat{\mathrm{g}}}(n)\tilde{\mathbf{x}}_{\hat{\mathbf{g}}}^T(n)$$
$$\mathbf{R}_{\hat{\mathbf{h}}}(n)=\lambda_{\hat{\mathrm{g}}}\mathbf{R}_{\hat{\mathbf{h}}}(n-1)+\tilde{\mathbf{x}}_{\hat{\mathbf{h}}}(n)\tilde{\mathbf{x}}_{\hat{\mathbf{h}}}^T(n)$$
$$\mathbf{p}_{\hat{\mathbf{g}}}(n)=\lambda_{\hat{\mathrm{h}}}\mathbf{p}_{\hat{\mathbf{g}}}(n-1)+\tilde{\mathbf{x}}_{\hat{\mathrm{g}}}(n)\mathrm{d}(n)$$
$$\mathbf{p}_{\hat{\mathbf{h}}}(n)=\lambda_{\hat{\mathrm{g}}}\mathbf{p}_{\hat{\mathbf{h}}}(n-1)+\tilde{\mathbf{x}}_{\hat{\mathbf{h}}}(n)\mathrm{d}(n)$$

, with

$$\tilde{\mathbf{x}}_{\hat{\mathbf{g}}}(n) = [\hat{\mathbf{g}}(n-1)\otimes\mathbf{I}_L]^T\tilde{\mathbf{x}}(n)$$
$$\tilde{\mathbf{x}}_{\hat{\mathbf{h}}}(n) = [\mathbf{I}_M\otimes\hat{\mathbf{h}}(n-1)]^T\tilde{\mathbf{x}}(n)$$

$\lambda_{\hat{\mathrm{h}}}$ $(0 \ll \lambda_{\hat{\mathrm{h}}} < 1)$

and

$\lambda_{\hat{\mathrm{g}}}$ $(0 \ll \lambda_{\hat{\mathrm{g}}} < 1)$

Forgetting factors

Matrix inversion lemma: $\mathbf{R}_{\hat{\mathbf{g}}}^{-1}(n)$, $\mathbf{R}_{\hat{\mathbf{h}}}^{-1}(n)$ → **RLS-BF** → *Complexity*: $O(L^2 + M^2)$

Auxiliary normal equations solvable with the ***Dichotomous Coordinate Descent*** algorithm: **RLS-DCD-BF**

RLS-DCD Algorithm for Bilinear Forms

Initialization:

$\widehat{\mathbf{h}}(0) = [\ 1 \quad 0 \quad \cdots \quad 0\ ]^T,\ \widehat{\mathbf{g}}(0) = \frac{1}{M}[\ 1 \quad 1 \quad \cdots \quad 1\ ]^T$

$\mathbf{R}_{\widehat{\mathbf{g}}}(0) = \delta \mathbf{I}_L,\ \mathbf{R}_{\widehat{\mathbf{h}}}(0) = \delta \mathbf{I}_M,\ \mathbf{r}_{\widehat{\mathbf{h}}}(0) = \mathbf{0}_{L\times 1},\ \mathbf{r}_{\widehat{\mathbf{g}}}(0) = \mathbf{0}_{M\times 1}$

For $n = 1, 2, \ldots$

Step 1: $\mathbf{R}_{\widehat{\mathbf{g}}}(n) = \lambda_{\widehat{\mathbf{h}}}\mathbf{R}_{\widehat{\mathbf{g}}}(n-1) + \widetilde{\mathbf{x}}_{\widehat{\mathbf{g}}}(n)\widetilde{\mathbf{x}}_{\widehat{\mathbf{g}}}^T(n)$

$\mathbf{R}_{\widehat{\mathbf{h}}}(n) = \lambda_{\widehat{\mathbf{g}}}\mathbf{R}_{\widehat{\mathbf{h}}}(n-1) + \widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}(n)\widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}^T(n)$

Step 2: $e(n) = d(n) - \widetilde{\mathbf{x}}_{\widehat{\mathbf{g}}}^T(n)\widehat{\mathbf{h}}(n-1) = d(n) - \widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}^T(n)\widehat{\mathbf{g}}(n-1)$

Step 3: $\widetilde{\mathbf{p}}_{\widehat{\mathbf{g}}}(n) = \lambda_{\widehat{\mathbf{h}}}\mathbf{r}_{\widehat{\mathbf{h}}}(n-1) + \widetilde{\mathbf{x}}_{\widehat{\mathbf{g}}}(n)e(n)$

$\widetilde{\mathbf{p}}_{\widehat{\mathbf{h}}}(n) = \lambda_{\widehat{\mathbf{g}}}\mathbf{r}_{\widehat{\mathbf{g}}}(n-1) + \widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}(n)e(n)$

Step 4: $\mathbf{R}_{\widehat{\mathbf{g}}}(n)\Delta\widehat{\mathbf{h}}(n) = \widetilde{\mathbf{p}}_{\widehat{\mathbf{g}}}(n) \xRightarrow{\text{DCD}} \Delta\widehat{\mathbf{h}}(n),\ \mathbf{r}_{\widehat{\mathbf{h}}}(n)$

$\mathbf{R}_{\widehat{\mathbf{h}}}(n)\Delta\widehat{\mathbf{g}}(n) = \widetilde{\mathbf{p}}_{\widehat{\mathbf{h}}}(n) \xRightarrow{\text{DCD}} \Delta\widehat{\mathbf{g}}(n),\ \mathbf{r}_{\widehat{\mathbf{g}}}(n)$

Step 5: $\widehat{\mathbf{h}}(n) = \widehat{\mathbf{h}}(n-1) + \Delta\widehat{\mathbf{h}}(n)$

$\widehat{\mathbf{g}}(n) = \widehat{\mathbf{g}}(n-1) + \Delta\widehat{\mathbf{g}}(n)$

RLS-DCD-BF-v1:

- $\tilde{\mathbf{x}}_{\hat{\mathrm{g}}}(n)$ ⇒ *time-shift property in the steady-state:* $\hat{\mathbf{g}}(n) \approx \hat{\mathbf{g}}(n-1)$
- $\mathbf{R}_{\hat{\mathrm{g}}}(n)$ *symmetric* ⇒ $\mathbf{R}_{\hat{\mathrm{g}}}^{(1)}(n) = \lambda_{\hat{\mathrm{h}}}\mathbf{R}_{\hat{\mathrm{g}}}^{(1)}(n-1) + \tilde{\mathbf{x}}_{\hat{\mathrm{g}}}(n)\tilde{\mathbf{x}}_{\hat{\mathrm{g}}}^{(1)}(n)$
- $(L-1)\times(L-1)$ *lower−right block of* $\mathbf{R}_{\hat{\mathrm{g}}}(\mathbf{n})$ $\approx$ $(L-1)\times(L-1)$ *upper−left block of* $\mathbf{R}_{\hat{\mathrm{g}}}(\mathbf{n-1})$
- *Complexity*: $\boldsymbol{O}(\boldsymbol{L}+\boldsymbol{M}^2)$

RLS-DCD-BF-v2:

- $\tilde{\mathbf{x}}_{\hat{\mathrm{g}}}(n)$ *and* $\tilde{\mathbf{x}}_{\hat{\mathrm{h}}}(n)$ *are independent and have the same power*
- *same approach for* $\mathbf{R}_{\hat{\mathrm{h}}}(\mathrm{n})$
- *Complexity*: $\boldsymbol{O}(\boldsymbol{L}+\boldsymbol{M})$

Regularized RLS Algorithm for Bilinear Forms

The cost functions (LS criterion):

$$J_{\hat{h}}[\hat{\mathbf{g}}(n)] = \sum_{i=1}^{n} \lambda_{\hat{g}}^{n-i} [d(i) - \hat{\mathbf{g}}^T(n)\tilde{\mathbf{x}}_{\hat{h}}(i)]^2 + \delta_{\hat{\mathbf{g}}}\|\hat{\mathbf{g}}(n)\|^2$$

$$J_{\hat{g}}[\hat{\mathbf{h}}(n)] = \sum_{i=1}^{n} \lambda_{\hat{h}}^{n-i} [d(i) - \hat{\mathbf{h}}^T(n)\tilde{\mathbf{x}}_{\hat{g}}(i)]^2 + \delta_{\hat{\mathbf{h}}}\|\hat{\mathbf{h}}(n)\|^2$$

, where $\lambda_{\hat{h}}\ (0 \ll \lambda_{\hat{h}} < 1)$ and $\lambda_{\hat{g}}\ (0 \ll \lambda_{\hat{g}} < 1)$

Forgetting factors

Regularization parameters

The updates:

$$\hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n-1) + [\mathbf{R}_{\hat{\mathbf{g}}}(n) + \delta_{\hat{\mathbf{h}}}\mathbf{I}_L]^{-1}\tilde{\mathbf{x}}_{\hat{\mathbf{g}}}(n)e(n)$$

$$\hat{\mathbf{g}}(n) = \hat{\mathbf{g}}(n-1) + [\mathbf{R}_{\hat{\mathbf{h}}}(n) + \delta_{\hat{\mathbf{g}}}\mathbf{I}_M]^{-1}\tilde{\mathbf{x}}_{\hat{\mathbf{h}}}(n)e(n)$$

, where

$$\tilde{\mathbf{x}}_{\hat{\mathbf{g}}}(n) = [\hat{\mathbf{g}}(n-1)\otimes\mathbf{I}_L]^T\tilde{\mathbf{x}}(n)$$

$$\tilde{\mathbf{x}}_{\hat{\mathbf{h}}}(n) = [\mathbf{I}_M\otimes\hat{\mathbf{h}}(n-1)]^T\tilde{\mathbf{x}}(n)$$

$$\mathbf{R}_{\hat{\mathbf{g}}}(n) = \lambda_{\hat{h}}\mathbf{R}_{\hat{\mathbf{g}}}(n-1) + \tilde{\mathbf{x}}_{\hat{g}}(n)\tilde{\mathbf{x}}_{\hat{\mathbf{g}}}^T(n)$$

$$\mathbf{R}_{\hat{\mathbf{h}}}(n) = \lambda_{\hat{g}}\mathbf{R}_{\hat{\mathbf{h}}}(n-1) + \tilde{\mathbf{x}}_{\hat{\mathbf{h}}}(n)\tilde{\mathbf{x}}_{\hat{\mathbf{h}}}^T(n)$$

Regularized RLS Algorithm for Bilinear Forms

$$\hat{\mathbf{h}}(n) = \mathbf{P}_{\hat{\mathbf{g}}}(n)\hat{\mathbf{h}}(n-1) + \tilde{\mathbf{h}}(n)$$
$$\hat{\mathbf{g}}(n) = \mathbf{P}_{\hat{\mathbf{h}}}(n)\hat{\mathbf{g}}(n-1) + \tilde{\mathbf{g}}(n)$$

The update equations can be rewritten as :

$$\mathbf{P}_{\hat{\mathbf{g}}}(n) = \mathbf{I}_L - \left[\mathbf{R}_{\hat{\mathbf{g}}}(n) + \delta_{\hat{\mathbf{h}}}\mathbf{I}_L\right]^{-1}\tilde{\mathbf{x}}_{\hat{\mathbf{g}}}(n)\tilde{\mathbf{x}}_{\hat{\mathbf{g}}}^T(n)$$
$$\mathbf{P}_{\hat{\mathbf{h}}}(n) = \mathbf{I}_M - \left[\mathbf{R}_{\hat{\mathbf{h}}}(n) + \delta_{\hat{\mathbf{g}}}\mathbf{I}_M\right]^{-1}\tilde{\mathbf{x}}_{\hat{\mathbf{h}}}(n)\tilde{\mathbf{x}}_{\hat{\mathbf{h}}}^T(n)$$

where

$$\tilde{\mathbf{h}}(n) = \left[\mathbf{R}_{\hat{\mathbf{g}}}(n) + \delta_{\hat{\mathbf{h}}}\mathbf{I}_L\right]^{-1}\tilde{\mathbf{x}}_{\hat{\mathbf{g}}}(n)d(n)$$
$$\tilde{\mathbf{g}}(n) = \left[\mathbf{R}_{\hat{\mathbf{h}}}(n) + \delta_{\hat{\mathbf{g}}}\mathbf{I}_M\right]^{-1}\tilde{\mathbf{x}}_{\hat{\mathbf{h}}}(n)d(n)$$

→ The correction components of the algorithm

- Let us define

$$\tilde{e}_{\hat{\mathbf{g}}}(n) = d(n) - \tilde{\mathbf{h}}^T(n)\tilde{\mathbf{x}}_{\hat{\mathbf{g}}}(n)$$
$$\tilde{e}_{\hat{\mathbf{h}}}(n) = d(n) - \tilde{\mathbf{g}}^T(n)\tilde{\mathbf{x}}_{\hat{\mathbf{h}}}(n)$$

- In the context of real-world system identification problems, the main purpose is to recover the noise signal from the error of the adaptive filter.
- We could find $\delta_{\hat{\mathbf{h}}}$ and $\delta_{\hat{\mathbf{g}}}$ in such a way that: $E\left[\tilde{e}_{\hat{\mathbf{g}}}^2(n)\right] = E\left[\tilde{e}_{\hat{\mathbf{h}}}^2(n)\right] = \sigma_w^2$

Regularized RLS Algorithm for Bilinear Forms

The quadratic equations:

$$\delta_{\hat{\mathbf{h}}}^2 - \frac{2\delta_{\hat{\mathbf{h}}} L\sigma_x^2 \nu_{\hat{\mathbf{g}}}}{\text{SNR}} - \frac{\left(L\sigma_x^2 \nu_{\hat{\mathbf{g}}}\right)^2}{\text{SNR}} = 0$$

$$\delta_{\hat{\mathbf{g}}}^2 - \frac{2\delta_{\hat{\mathbf{g}}} M\sigma_x^2 \nu_{\hat{\mathbf{h}}}}{\text{SNR}} - \frac{\left(M\sigma_x^2 \nu_{\hat{\mathbf{h}}}\right)^2}{\text{SNR}} = 0$$

, where

$$\nu_{\hat{\mathbf{g}}} = E\left[\|\hat{\mathbf{g}}(n-1)\|\right]^2$$

$$\nu_{\hat{\mathbf{h}}} = E\left[\|\hat{\mathbf{h}}(n-1)\|\right]^2$$

- The obvious solutions of these equations lead to the regularization parameters:

$$\delta_{\hat{\mathbf{h}}} = \frac{LE\left[\|\hat{\mathbf{g}}(n-1)\|^2\right]\left(1+\sqrt{1+\text{SNR}}\right)}{\text{SNR}}\sigma_{\boldsymbol{x}}^2$$

$$\delta_{\hat{\mathbf{g}}} = \frac{ME\left[\|\hat{\mathbf{h}}(n-1)\|^2\right]\left(1+\sqrt{1+\text{SNR}}\right)}{\text{SNR}}\sigma_{\boldsymbol{x}}^2$$

- Let us assume that the adaptive filter has converged to a certain degree: $\sigma_y^2 \approx \sigma_{\hat{y}}^2$

- We can express the signal model in terms of power estimates: $\sigma_d^2 = \sigma_y^2 + \sigma_w^2$

Variable-Regularized RLS Algorithm for Bilinear Forms

- The power estimates can be evaluated in a recursive manner as:

$$\hat{\sigma}_d^2(n) = \gamma\hat{\sigma}_d^2(n-1) + (1-\gamma)d^2(n)$$
$$\hat{\sigma}_{\hat{y}}^2(n) = \gamma\hat{\sigma}_{\hat{y}}^2(n-1) + (1-\gamma)\hat{y}^2(n)$$

$0 \ll \gamma < 1$ →

$$\widehat{\mathrm{SNR}}(n) = \frac{\hat{\sigma}_{\hat{y}}^2(n)}{\left|\hat{\sigma}_d^2(n) - \hat{\sigma}_{\hat{y}}^2(n)\right|}$$

- The variable regularization parameters results in:

$$\delta_{\hat{\mathbf{h}}}(n) = L\|\hat{\mathbf{g}}(n-1)\|^2 s(n)\sigma_x^2$$
$$\delta_{\hat{\mathbf{g}}}(n) = M\left\|\hat{\mathbf{h}}(n-1)\right\|^2 s(n)\sigma_x^2$$

, where $s(n) = \frac{1+\sqrt{1+\widehat{\mathrm{SNR}}(n)}}{\widehat{\mathrm{SNR}}(n)}$ →

VR-RLS-BF

Complexity: $\boldsymbol{O}(\boldsymbol{L}^2 + \boldsymbol{M}^2)$

- The problem can be interpreted again in terms of solving the normal equations:

$$\underline{\mathbf{R}}_{\hat{\mathbf{g}}}(n)\hat{\mathbf{h}}(n) = \mathbf{p}_{\hat{\mathbf{g}}}(n)$$
$$\underline{\mathbf{R}}_{\hat{\mathbf{h}}}(n)\hat{\mathbf{g}}(n) = \mathbf{p}_{\hat{\mathbf{h}}}(n)$$

, where

$$\underline{\mathbf{R}}_{\hat{\mathbf{g}}}(n) = \widehat{\mathbf{R}}_{\hat{\mathbf{g}}}(n) + \delta_{\hat{\mathbf{h}}}(n)\mathbf{I}_L$$
$$\underline{\mathbf{R}}_{\hat{\mathbf{h}}}(n) = \widehat{\mathbf{R}}_{\hat{\mathbf{h}}}(n) + \delta_{\hat{\mathbf{g}}}(n)\mathbf{I}_M$$

and $\mathbf{p}_{\hat{\mathbf{g}}}(n)$ and $\mathbf{p}_{\hat{\mathbf{h}}}(n)$ as for **RLS-BF**

→ Auxiliary normal equations solvable with the ***Dichotomous Coordinate Descent*** algorithm: **VR-RLS-DCD-BF**

Variable-Regularized RLS Algorithm for Bilinear Forms

Initialization:

$\widehat{\mathbf{h}}(0) = [\ 1 \quad 0 \quad \cdots \quad 0\ ]^T, \ \widehat{\mathbf{g}}(0) = \frac{1}{M}[\ 1 \quad 1 \quad \cdots \quad 1\ ]^T$

$\mathbf{R}_{\widehat{\mathbf{g}}}(0) = \mathbf{0}_{L\times L}, \ \mathbf{R}_{\widehat{\mathbf{h}}}(0) = \mathbf{0}_{M\times M}$

$\underline{\mathbf{r}}_{\widehat{\mathbf{h}}}(0) = \mathbf{0}_{L\times 1}, \ \underline{\mathbf{r}}_{\widehat{\mathbf{g}}}(0) = \mathbf{0}_{M\times 1}$

For $n = 1, 2, \ldots$

Step 1: $\mathbf{R}_{\widehat{\mathbf{g}}}(n) = \lambda_{\widehat{\mathbf{h}}}\mathbf{R}_{\widehat{\mathbf{g}}}(n-1) + \widetilde{\mathbf{x}}_{\widehat{\mathbf{g}}}(n)\widetilde{\mathbf{x}}_{\widehat{\mathbf{g}}}^T(n)$

$\mathbf{R}_{\widehat{\mathbf{h}}}(n) = \lambda_{\widehat{\mathbf{g}}}\mathbf{R}_{\widehat{\mathbf{h}}}(n-1) + \widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}(n)\widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}^T(n)$

Step 2: Compute $\delta_{\widehat{\mathbf{h}}}(n)$ and $\delta_{\widehat{\mathbf{g}}}(n)$ using (23)–(24)

Step 3: $\underline{\mathbf{R}}_{\widehat{\mathbf{g}}}(n) = \mathbf{R}_{\widehat{\mathbf{g}}}(n) + \delta_{\widehat{\mathbf{h}}}(n)\mathbf{I}_L$

$\underline{\mathbf{R}}_{\widehat{\mathbf{h}}}(n) = \mathbf{R}_{\widehat{\mathbf{h}}}(n) + \delta_{\widehat{\mathbf{g}}}(n)\mathbf{I}_M$

Step 4: $e(n) = d(n) - \widetilde{\mathbf{x}}_{\widehat{\mathbf{g}}}^T(n)\widehat{\mathbf{h}}(n-1) = d(n) - \widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}^T(n)\widehat{\mathbf{g}}(n-1)$

Step 5: $\underline{\widetilde{\mathbf{p}}}_{\widehat{\mathbf{g}}}(n) = \lambda_{\widehat{\mathbf{h}}}\underline{\mathbf{r}}_{\widehat{\mathbf{h}}}(n-1) + \widetilde{\mathbf{x}}_{\widehat{\mathbf{g}}}(n)e(n)$

$\underline{\widetilde{\mathbf{p}}}_{\widehat{\mathbf{h}}}(n) = \lambda_{\widehat{\mathbf{g}}}\underline{\mathbf{r}}_{\widehat{\mathbf{g}}}(n-1) + \widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}(n)e(n)$

Step 6: $\underline{\mathbf{R}}_{\widehat{\mathbf{g}}}(n)\Delta\widehat{\mathbf{h}}(n) = \underline{\widetilde{\mathbf{p}}}_{\widehat{\mathbf{g}}}(n) \overset{\text{DCD}}{\Longrightarrow} \Delta\widehat{\mathbf{h}}(n), \ \underline{\mathbf{r}}_{\widehat{\mathbf{h}}}(n)$

$\underline{\mathbf{R}}_{\widehat{\mathbf{h}}}(n)\Delta\widehat{\mathbf{g}}(n) = \underline{\widetilde{\mathbf{p}}}_{\widehat{\mathbf{h}}}(n) \overset{\text{DCD}}{\Longrightarrow} \Delta\widehat{\mathbf{g}}(n), \ \underline{\mathbf{r}}_{\widehat{\mathbf{g}}}(n)$

Step 7: $\widehat{\mathbf{h}}(n) = \widehat{\mathbf{h}}(n-1) + \Delta\widehat{\mathbf{h}}(n)$

$\widehat{\mathbf{g}}(n) = \widehat{\mathbf{g}}(n-1) + \Delta\widehat{\mathbf{g}}(n)$

VR-RLS-DCD-BF-v1:

- $\tilde{\mathbf{x}}_{\hat{\mathrm{g}}}(n)$ ⇨ *time-shift property in the steady-state:* $\hat{\mathbf{g}}(n) \approx \hat{\mathbf{g}}(n-1)$
- $\mathbf{R}_{\hat{\mathrm{g}}}(n)$ *symmetric* ⇨ $\mathbf{R}_{\hat{\mathrm{g}}}^{(1)}(n) = \lambda_{\hat{\mathrm{h}}}\mathbf{R}_{\hat{\mathrm{g}}}^{(1)}(n-1) + \tilde{\mathbf{x}}_{\hat{\mathrm{g}}}(n)\tilde{\mathbf{x}}_{\hat{\mathrm{g}}}^{(1)}(n)$
- $(\mathrm{L}-1)\times(\mathrm{L}-1)$ *lower–right block of* $\mathbf{R}_{\hat{\mathbf{g}}}(\mathbf{n})$ $\approx$ $(\mathrm{L}-1)\times(\mathrm{L}-1)$ *upper–left block of* $\mathbf{R}_{\hat{\mathbf{g}}}(\mathbf{n-1})$
- *Complexity:* $\boldsymbol{O}(\boldsymbol{L}+\boldsymbol{M}^2)$

VR-RLS-DCD-BF-v2:

- $\tilde{\mathbf{x}}_{\hat{\mathrm{g}}}(n)$ *and* $\tilde{\mathbf{x}}_{\hat{\mathrm{h}}}(n)$ *are independent and have the same power*
- *same approach for* $\mathbf{R}_{\hat{\mathbf{h}}}(\mathrm{n})$
- *Complexity:* $\boldsymbol{O}(\boldsymbol{L}+\boldsymbol{M})$

Simulation Results

- **Conditions:**
  - → system identification, L = 64, M = 8
  - → h, g – randomly generated (Gaussian distribution)
  - → input signals – AR(1) processes; each one is generated by filtering a white Gaussian noise through a first-order system with the transfer function $1/(1-0.8z^{-1})$
  - → only one successful DCD iteration used
  - → additive noise w(n) – WGN
  - → $\lambda_{\widehat{\mathrm{h}}} = \lambda_{\widehat{\mathrm{g}}} = 1 - 1/(2ML)$
  - → measure of performance:

$$\mathrm{NPM}\left[\mathbf{f}, \hat{\mathbf{f}}(n)\right] = 1 - \left[\frac{\mathbf{f}^T \hat{\mathbf{f}}(n)}{\|\mathbf{f}\| \, \left\|\hat{\mathbf{f}}(n)\right\|}\right]^2 \; [\mathrm{dB}]$$

- **Algorithms:**
  - → RLS-DCD-BF
  - → VR-RLS-BF
  - → VR-RLS-DCD-BF

Simulation Results

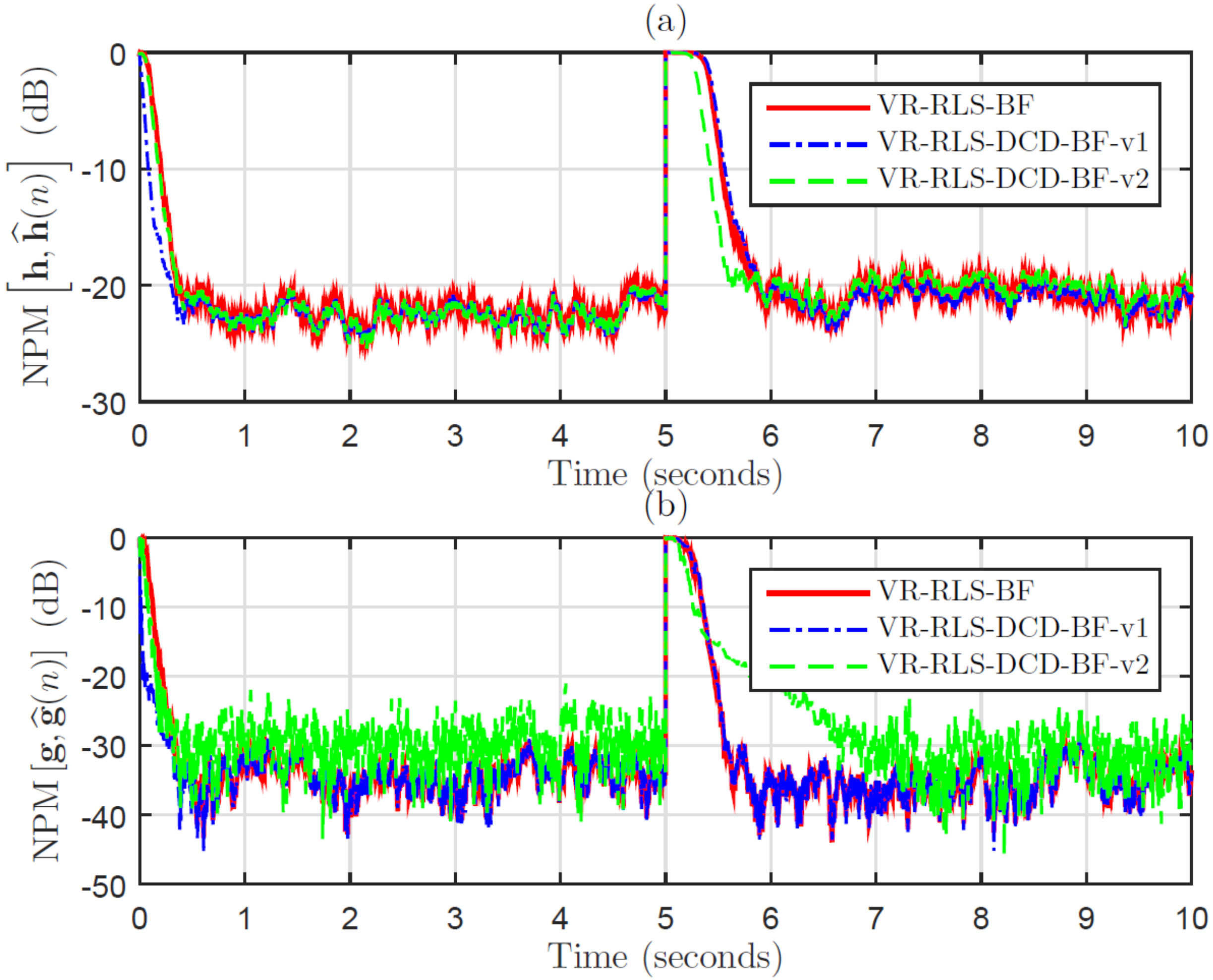

Figure 1. Comparison of the VR-based algorithms in terms of (a) NPM$\left[\mathbf{h}, \hat{\mathbf{h}}(n)\right]$ and (b) NPM$[\mathbf{g}, \hat{\mathbf{g}}(n)]$. The system changes after 5 seconds. The input signals are AR(1) processes and SNR $=$ 10 dB.

Simulation Results

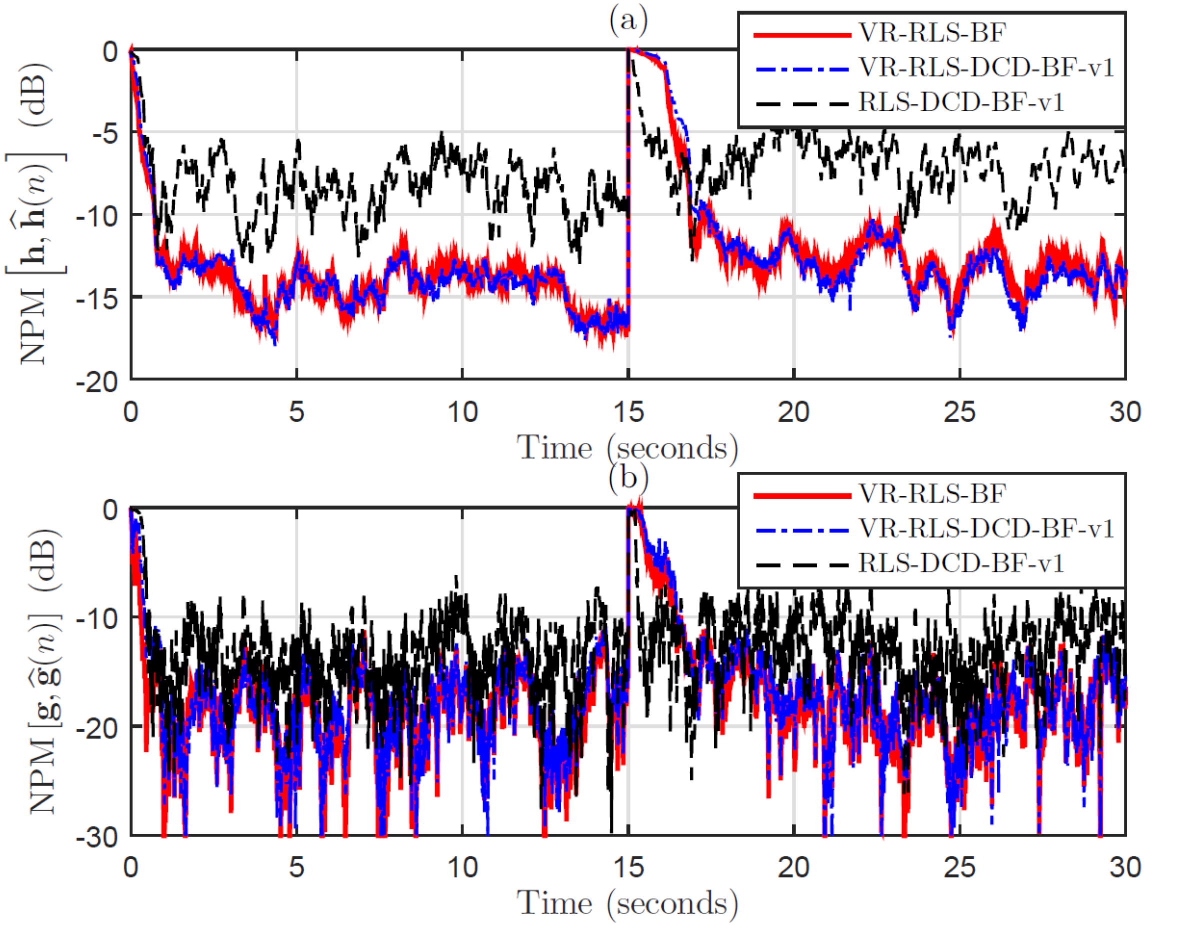

Figure 2. Comparison of the VR-based algorithms in terms of (a) NPM$\left[\mathbf{h}, \hat{\mathbf{h}}(n)\right]$ and (b) NPM$[\mathbf{g}, \hat{\mathbf{g}}(n)]$. The system changes after 15 seconds. The input signals are speech sequences and SNR $=0$ dB.

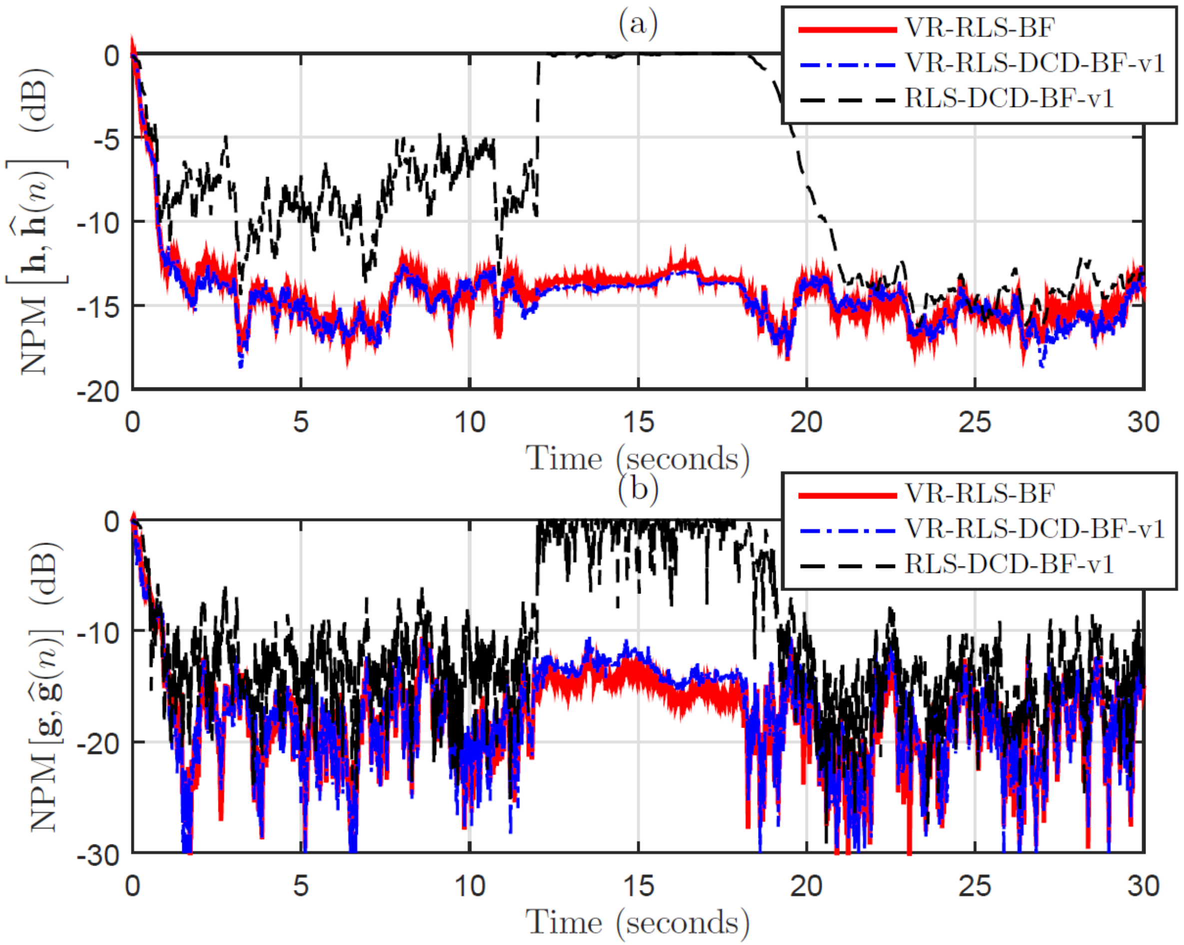

Figure 3. Comparison of the VR-RLS-BF, VR-RLS-DCD-BF-v1, and RLSDCD-v1 algorithms in terms of (a) NPM$\left[\mathbf{h}, \hat{\mathbf{h}}(n)\right]$ and (b) NPM$[\mathbf{g}, \hat{\mathbf{g}}(n)]$. The SNR decreases from system 0 dB to -25 dB between times 12 and 18 seconds.

Conclusions

- We focused on the regularization terms of the RLS algorithm tailored for the identification of bilinear forms.
- The bilinear form was defined with respect to the impulse responses.
- We have presented a method to find the regularization parameters depending on the SNR.
- Using a proper estimation of the SNR, a variable-regularized solution was proposed – VR-RLS-BF, together with two low-complexity versions based on the DCD method.
- Simulations have shown that the VR-based algorithms outperform their non-regularized counterpart, mainly in terms of robustness against SNR variations.
- Future works will focus on the extension of these solutions in case of multilinear forms, by exploiting tensor-based adaptive algorithms. In this context, the decomposition methods can be combined with low-rank approximations, aiming the identification of more general forms of impulse responses.

Thank you for your attention!

This work was supported by a grant of the Romanian Ministry of Education and Research, CNCS-UEFISCDI, project number PN-III-P1-1.1-TE-2019-0529, within PNCDI III