A Data-Reuse Approach for the RLS-DCD Algorithm

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

University Politehnica of Bucharest, Romania
ionut.ficiu22@gmail.com, cristian@comm.pub.ro, camelia.elisei@romatsa.ro, canghel@comm.pub.ro, pale@comm.pub.ro*

** Presenter*

Presenter’s Biography

Ionut Fîciu obtained his bachelor’s degree in 2018 - the Systems and Technologies in Telecommunications specialization, University Politehnica of Bucharest (UPB), Romania – valedictorian. He is a PhD. student in the Telecommunications Department at UPB. His main research topic is related to the field of adaptive systems.

He works as a member in three research grants, and during his first 18 months of doctoral studies he published more than ten papers (including in Q2/Q3 journals).

Summary

- Introduction
- System Model
- The Recursive Least Squares - Dichotomous Coordinate Descent (RLS-DCD) Algorithm
- Data-Reuse (DR) RLS-DCD Algorithm
- Simulation Results
- Conclusions

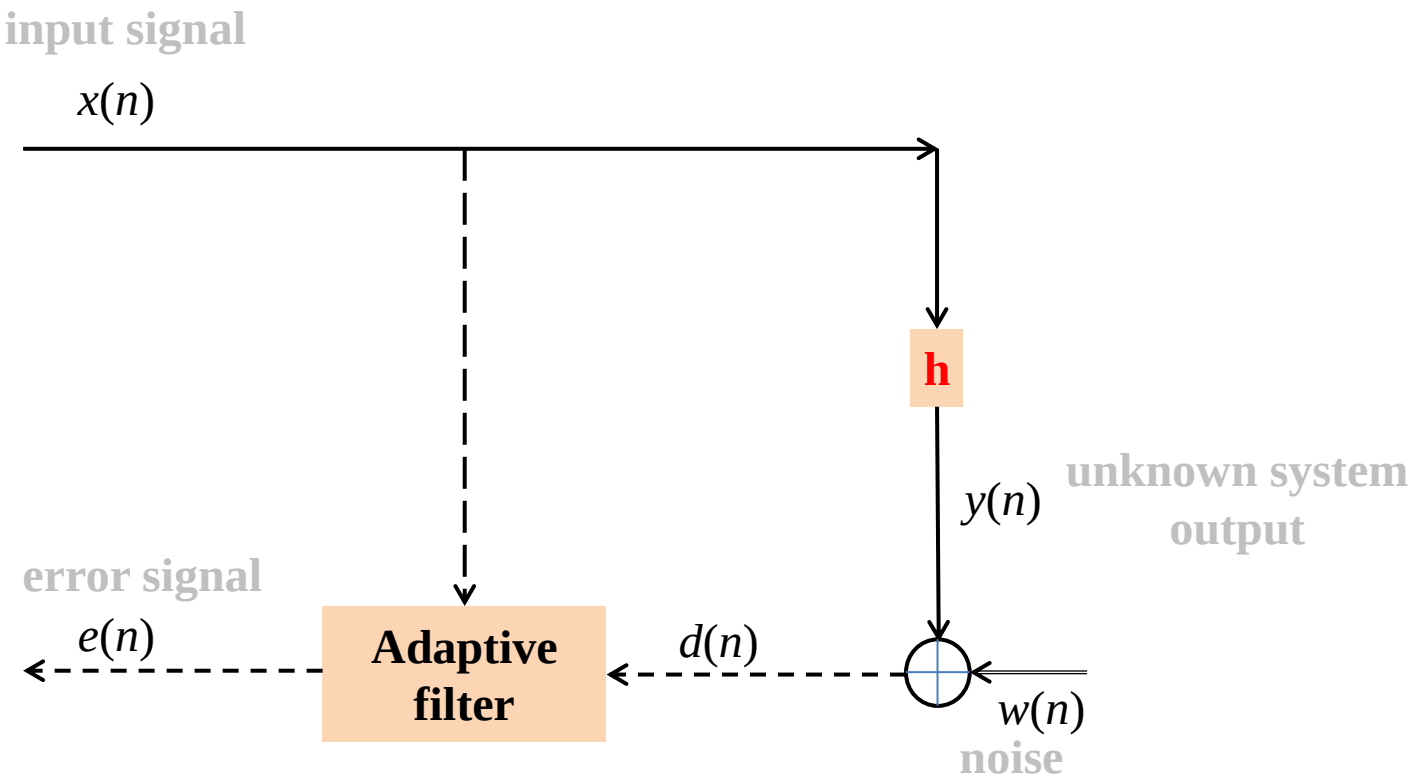
INTRODUCTION

- RLS adaptive algorithm:
  - Mitigation of highly correlated input signals
  - High arithmetic costs

- The RLS-DCD method:
  - The RLS algorithm => the matrix inversion problem
    - » The Dichotomous Coordinate Descent (DCD) iterations

- The Data-Reuse (DR) RLS-DCD:
  - Increased tracking/convergence speed
    - » The use of the same input information at time index *n*
    - » Repeat the update process (*reuse*) the **same data set** N_{it} times

System Model



$$\mathbf{x}(n) = \begin{bmatrix} x(n) & x(n-1) & \dots & x\ (n-L+1) \end{bmatrix}^T$$

$$\mathbf{h} = \begin{bmatrix} h_0 & h_1 & \dots & h_{L-1} \end{bmatrix}^T$$

$$d(n) = y(n) + w(n) = \mathbf{h}^T\mathbf{x}(n) + w(n) \quad \dashrightarrow \quad e(n) = d(n) - \hat{y}(n) = d(n) - \hat{\mathbf{h}}(n-1)\mathbf{x}(n)$$

RLS-DCD Algorithm

- The least-squares (LS) error criterion:

$$J\left[\hat{\mathbf{h}}(n)\right] = \sum_{i=1}^{n} \lambda^{n-i} \left| d(i) - \hat{\mathbf{h}}^T(n)\mathbf{x}(i) \right|^2 \quad \longrightarrow \quad 0 < \lambda < 1$$

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

where

$$\mathbf{R}(n) = \sum_{i=1}^{n} \lambda^{n-i} \mathbf{x}(i)\mathbf{x}^T(i) \quad \longrightarrow \quad L \text{ x } L$$

$$\mathbf{p}(n) = \sum_{i=1}^{n} \lambda^{n-i} \mathbf{x}(i)d(i) \quad \longrightarrow \quad L \text{ x } 1$$

- Matrix inversion:
  - RLS $\longrightarrow \sim L^2$
  - FRLS $\longrightarrow \sim L$

$\longrightarrow$ RLS-DCD

RLS-DCD Algorithm

$0: \quad \hat{\mathbf{h}}(0) = \mathbf{0}, \mathbf{r}(0) = \mathbf{0}, \mathbf{R}(0) = \delta \mathbf{I}_L$

For $n = 1, 2 \ldots$

$1: \quad \mathbf{R}(n) = \lambda \mathbf{R}(n-1) + \mathbf{x}(n)\mathbf{x}^T(n)$

$2: \quad e(n) = d(n) - \hat{\mathbf{h}}^T(n-1)\mathbf{x}(n)$

$3: \quad \mathbf{p}_0(n) = \lambda \mathbf{r}(n-1) + \mathbf{x}(n)e(n)$

$= \lambda \mathbf{r}(n-1) + \mathbf{r}_{e,x}(n)$

$4: \quad \mathbf{R}(n)\Delta\hat{\mathbf{h}}(n) = \mathbf{p}_0(n) \Rightarrow \Delta\hat{\mathbf{h}}(n), \mathbf{r}(n)$

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

Only the first columns has to be computed

$\mathbf{R}^{(1)}(n) = \lambda \mathbf{R}^{(1)}(n-1) + x(n)\mathbf{x}(n)$

- Solved with Dichotomous Coordinate Descent (DCD):
  - Uses only additions
  - Important parameters:
    - » N_u (maximum number of allowed updates/ "successful iterations")
    - » M_b (number of bits used for the representation of the solution vector)
  - Arithmetic complexity ~ LN_u, $N_u << L$

DR-RLS-DCD Algorithm

- At each time index "n", N_{it}:

The regular error expression; use results from time index n-1

$$e(n) = \begin{cases} d(n) - \hat{\mathbf{h}}^T(n-1)\mathbf{x}(n) = e_0(n), & k = 0 \\ e_{k-1}(n) - \Delta\hat{\mathbf{h}}_{k-1}^T(n)\mathbf{x}(n), & k = 1 \ldots N_{it} - 1 \end{cases}$$

Reuse of the same input information (for N_{it}-1 *times*) ~ $N_{it}N_u$

Update the ***residual component***: use results from time index n-1

$$\mathbf{r}_k(n) = \begin{cases} \lambda\mathbf{r}_{N_{it}-1}(n-1) + e_0(n)\mathbf{x}(n), & k = 0 \\ \mathbf{p}_k(n) = \mathbf{r}_{k-1}(n) + e_k(n)\mathbf{x}(n), & k = 1 \ldots N_{it} - 1 \end{cases}$$

Update the ***residual component*** for the ***reuse*** process. ~ L

DR-RLS-DCD Algorithm

$0: \quad \hat{\mathbf{h}}(0) = \mathbf{0}, \mathbf{r}(0) = \mathbf{0}, \mathbf{R}(0) = \delta \mathbf{I}_L$

For $n = 1, 2 \ldots$

$1: \quad \mathbf{R}(n) = \lambda \mathbf{R}(n-1) + \mathbf{x}(n)\mathbf{x}^T(n)$

$2: \quad e(n) = d(n) - \hat{\mathbf{h}}^T(n-1)\mathbf{x}(n)$

$3: \quad \mathbf{p}_0(n) = \lambda \mathbf{r}(n-1) + \mathbf{x}(n)e(n)$

$= \lambda \mathbf{r}(n-1) + \mathbf{r}_{e,x}(n)$

$4: \quad \mathbf{R}(n)\Delta\hat{\mathbf{h}}(n) = \mathbf{p}(n) \Rightarrow \Delta\hat{\mathbf{h}}(n), \mathbf{r}(n$

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

- Run the DCD for N_{it} times
- Low complexity => acceptable cost
- Overall complexity ~ $(N_{it}\text{-}1)N_u L$

$0: \quad \hat{\mathbf{h}}(0) = \mathbf{0}, \mathbf{r}(0) = \mathbf{0}, \mathbf{R}(0) = \delta \mathbf{I}_L$

For $n = 1, 2 \ldots$

$1: \quad \mathbf{R}(n) = \lambda \mathbf{R}(n-1) + \mathbf{x}(n)\mathbf{x}^T(n)$

$2: \quad e(n) = d(n) - \hat{\mathbf{h}}^T(n-1)\mathbf{x}(n)$

3: For $k = 1, 2 \ldots N_{it}$

$a) \quad$ Compute $e_k(n)$

$b) \quad$ Compute $\mathbf{p}_k(n)$

$c) \quad \mathbf{R}(n)\Delta\hat{\mathbf{h}}_k(n) = \mathbf{p}_k(n) \Rightarrow \Delta\hat{\mathbf{h}}_k(n), \mathbf{r}_k(n)$

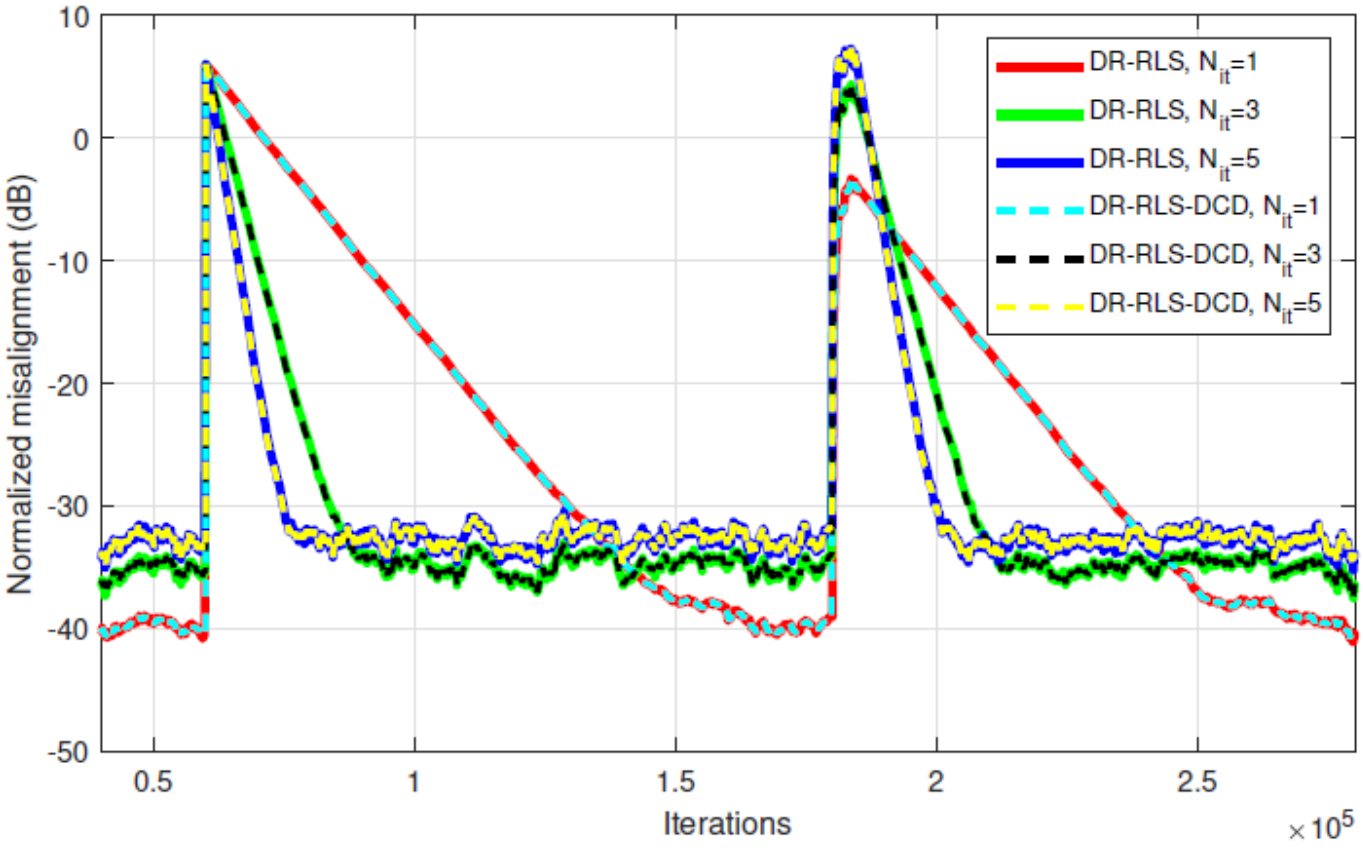
$d) \quad \hat{\mathbf{h}}_k(n) = \hat{\mathbf{h}}_k(n-1) + \Delta\hat{\mathbf{h}}_k(n)$

Simulation Results

- Configuration: system identification (tracking) + low SNR conditions
- DR-RLS vs DR-RLS-DCD

- Unknown system:
  - Fourth impulse response from the G.168 ITU-T Recommendation (L=128)
- Input signal:
  - AR(1) sequence with pole 0.9
- Forgetting factor: $\lambda = 1 - 1/(KL)$; $K = 128$
- DCD: $N_u = 4$, $M_b = 16$, $H = 1$
- SNR = 20 dB
- Normalized misalignment: $20\log_{10}\left[\left\|\mathbf{h}-\hat{\mathbf{h}}(n)\right\|_2 \Big/ \left\|\mathbf{h}\right\|_2\right]$ (dB)

Simulation Results



Performance of the DR-RLS and DR-RLS-DCD algorithms for different values of N_{it}. The unknown system changes at time index 60001, and the SNR is decreased for 5000 iterations, starting with index 180001.

Conclusions

- DR-RLS-DCD: compromise between tracking speed and accuracy at steady-state.
- DR-RLS-DCD matches the performance of DR-RLS.
- Arithmetic workload ~ proportional to a small multiple of L ➔ $(N_{it}\text{-}1)N_uL$
  - Attractive for hardware implementations
- Future research:
  - DR-RLS-DCD with variable N_{it}
  - *Regularized/Variable-Regularized* DR-RLS-DCD versions

Thank you !

Please refer any questions to:

ionut.ficiu22@gmail.com
cristian@comm.pub.ro
camelia.elisei@romatsa.ro
canghel@comm.pub.ro
pale@comm.pub.ro