A PROPORTIONATE AFFINE PROJECTION ALGORITHM USING FAST RECURSIVE FILTERING AND DICHOTOMOUS COORDINATE DESCENT ITERATIONS

Felix Albu

Valahia University of Targoviste

Outline

- Motivation and objectives
- Development of the algorithm
- Simulation results
- Conclusions

Motivation and objectives

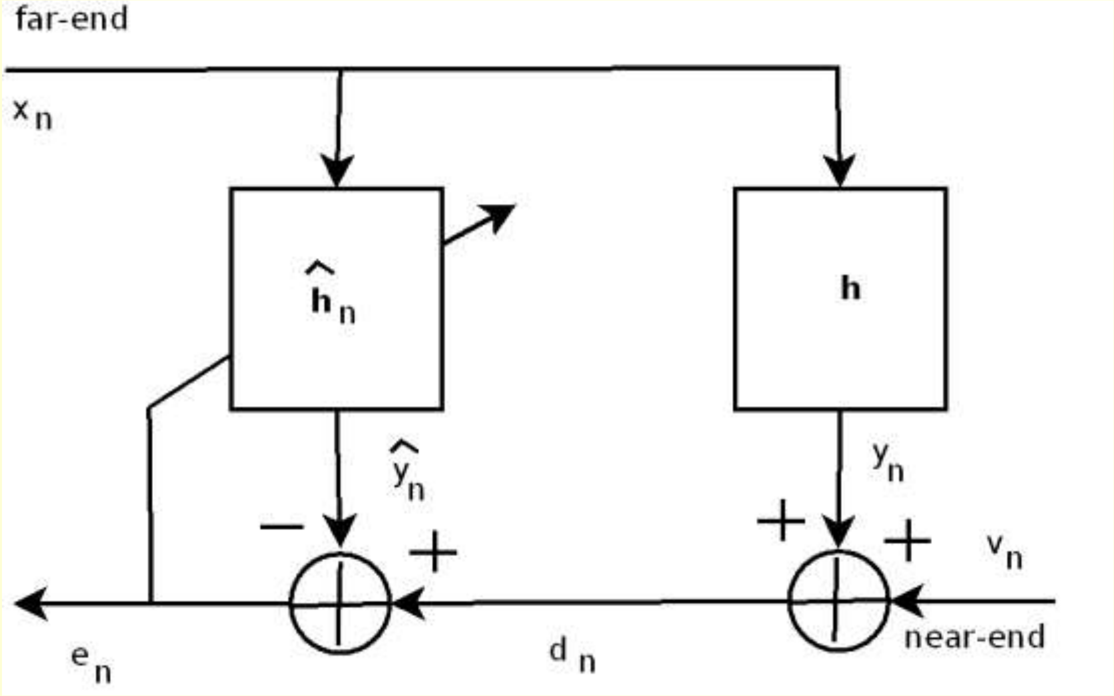
- The proportionate affine projection algorithm (PAPA) has a good convergence speed and low computational complexity. It is well known that it has superior performance to APA.

- Recently, two proportionate-type APA called MIPAPA was developed, taking into account the "history" of the proportionate factors.
- It was shown that they have better performance than IPAPA

- Objectives:
  - To obtain an efficient PAPA
  - To validate its performance and compare it with other algorithms
  - To identify the strengths and weaknesses of the algorithm

Development of the algorithm

- The far-end signal goes through the echo path **h**, providing the echo signal. The echo signal is added with the near-end signal (which can contain both the background noise and the near-end speech), resulting the microphone signal. The adaptive filter aims to produce at its output an estimate of the echo, while the error signal should contain an estimate of the near-end signal.



APA

$\mathbf{y}(n) = \mathbf{X}^T(n)\hat{\mathbf{h}}(n-1)$	(1)
$\mathbf{e}(n) = \mathbf{d}(n) - \mathbf{y}(n)$	(2)
$\hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n-1) + \mu\mathbf{X}(n)\left[\delta\mathbf{I}_p + \mathbf{X}^T(n)\mathbf{X}(n)\right]^{-1}\mathbf{e}(n)$	(3)

IPAPA

$\mathbf{y}(n) = \mathbf{X}^T(n)\hat{\mathbf{h}}(n-1)$	(1)
$\mathbf{e}(n) = \mathbf{d}(n) - \mathbf{y}(n)$	(2)
$g_l(n-1) = \frac{1-\alpha}{2L} + (1+\alpha)\frac{\left\|\hat{h}_l(n-1)\right\|}{2\sum_{i=0}^{L-1}\left\|\hat{h}_i(n-1)\right\| + \varepsilon}$	(3)
$\mathbf{P}(n) = \mathbf{G}(n)\mathbf{X}(n)$	(4)
$\hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n-1) + \mu\mathbf{P}(n)\left[\delta\mathbf{I}_p + \mathbf{X}^T(n)\mathbf{P}(n)\right]^{-1}\mathbf{e}(n)$	(5)

FMIPAPA

$\mathbf{P}'(n) = \left[\mathbf{g}(n-1) \odot \mathbf{x}(n) \quad \mathbf{P}'_{-1}(n-1)\right]$	(1)
$\mathbf{P}'_{-1}(n-1) = \left[\mathbf{g}(n-2) \odot \mathbf{x}(n-1) \quad \mathbf{g}(n-p) \odot \mathbf{x}(n-p+1)\right]$	(2)
$\mathbf{y}(n-1) = \mathbf{X}^T(n-1)\hat{\mathbf{h}}(n-2) = \left[\mathbf{x}^T(n-1)\hat{\mathbf{h}}(n-2) \ \ldots \ \mathbf{x}^T(n-p)\hat{\mathbf{h}}(n-2)\right]^T = \left[y^0(n-1) \ y^1(n-1) \ \ldots \ y^{p-1}(n-1)\right]^T$	(3)
$\mathbf{y}(n) = \mathbf{X}^T(n)\hat{\mathbf{h}}(n-1) = \mathbf{z}(n) + \mathbf{X}^T(n)\mathbf{P}'(n-1)\hat{\boldsymbol{\varepsilon}}(n-1)$	(4)
$\mathbf{z}(n) = \mathbf{X}^T(n)\hat{\mathbf{h}}(n-2) = \left[\mathbf{x}^T(n)\hat{\mathbf{h}}(n-2) \ldots \mathbf{x}^T(n-p+1)\hat{\mathbf{h}}(n-2)\right]^T = \left[\mathbf{x}^T(n)\hat{\mathbf{h}}(n-2) \ y^0(n-1) \ \ldots \ y^{p-2}(n-1)\right]^T$	(5)

FMIPAPA-DCD

$\hat{\boldsymbol{\varepsilon}}(-1)=0, \hat{\mathbf{h}}(-1)=0, \quad \mathbf{x}(-1)=\mathbf{0}, \mathbf{P}'(-1)=\mathbf{0}$	(1)
$\mathbf{z}(n)=\left[\mathbf{x}^T(n)\hat{\mathbf{h}}(n-2)\ y^0(n-1)\ \ldots y^{p-2}(n-1)\right]^T$	(2)
$\mathbf{F}(n)=\mathbf{X}^T(n)\mathbf{P}'(n-1)$	(3)
$\mathbf{y}(n)=\mathbf{z}(n)+\mathbf{F}(n)\hat{\boldsymbol{\varepsilon}}(n-1)$	(4)
$\mathbf{e}(n)=\mathbf{d}(n)-\mathbf{y}(n)$	(5)
$\mathbf{P}'(n)=\left[\mathbf{g}(n-1)\odot\mathbf{x}(n)\ \ \mathbf{P}'_{-1}(n-1)\right]$	(6)
$\mathbf{S}(n)=\delta\mathbf{I}_p+\mathbf{X}^T(n)\mathbf{P}'(n)$	(7)
Solve $\mathbf{S}(n)\boldsymbol{\varepsilon}(n)=\mathbf{e}(n)$ using the DCD method	(8)
$\hat{\mathbf{h}}(n)=\hat{\mathbf{h}}(n-1)+\mu\mathbf{P}'(n)\hat{\boldsymbol{\varepsilon}}(n)$	(9)

Development of the algorithm

System to solve: $\mathbf{R}\boldsymbol{\varepsilon} = \mathbf{e}$

Initialization: $\boldsymbol{\varepsilon} = 0,\ d = H,\ q = 0$

For $m = 1 : M_b$

$d = d/2$

(a) $\text{flag} = 0$

For $p = 0 : N-1$

if $\left|e_p\right| > (d/2)[\mathbf{R}]_{p,p}$, *then*

$flag = 1, q = q + 1$

$\varepsilon_p = \varepsilon_p + \text{sgn}\left(e_p\right) \cdot d$

$\mathbf{e} = \mathbf{e} - \text{sgn}\left(e_p\right) \cdot d \cdot \mathbf{R}(:, p)$

if $q > N_u$, then the algorithm stops

End of the p -loop

If $flag = 1$, then go to (a)

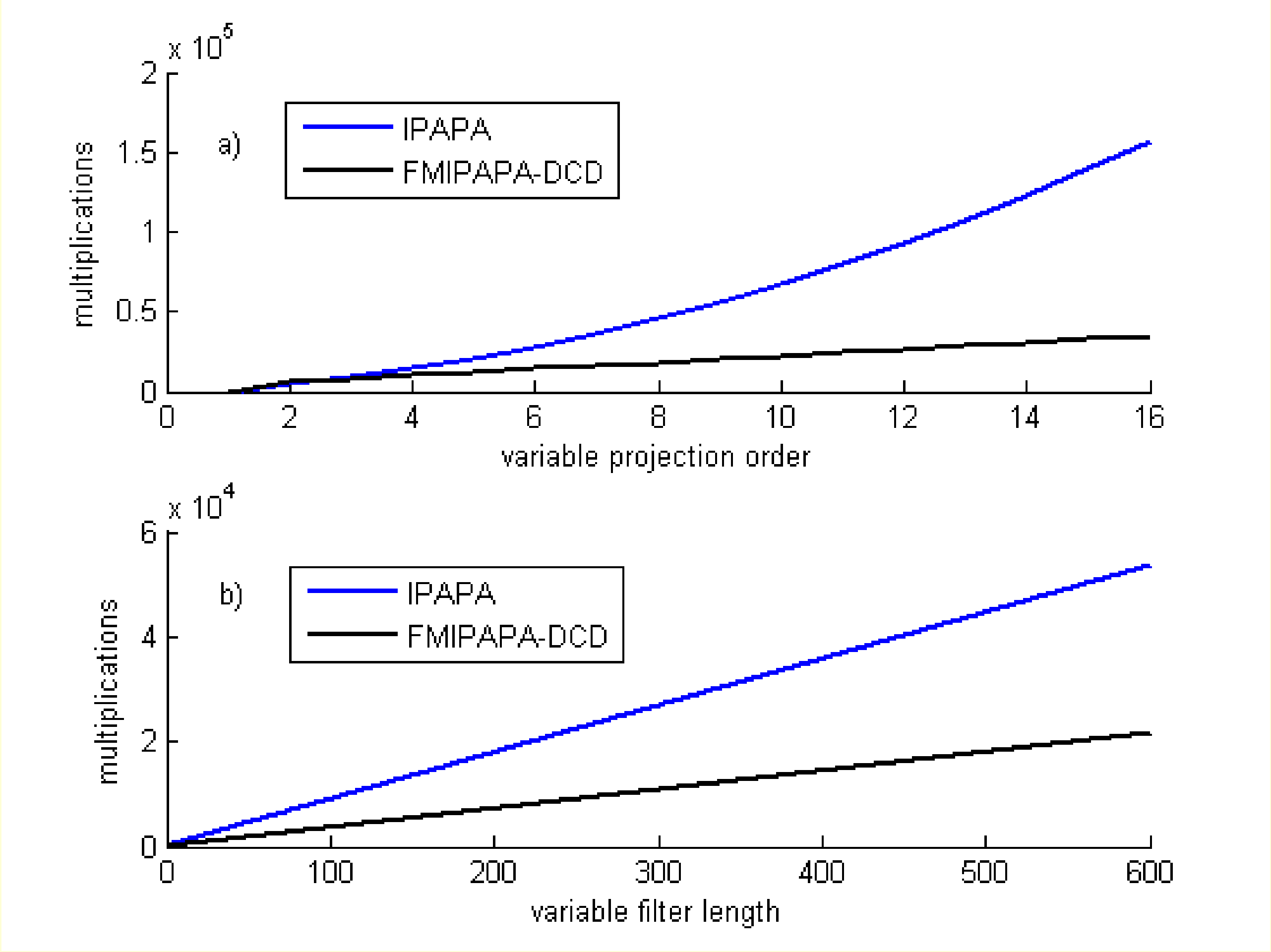
End of the m -loop

The dichotomous coordinate descent algorithm (DCD)

Computational Complexity

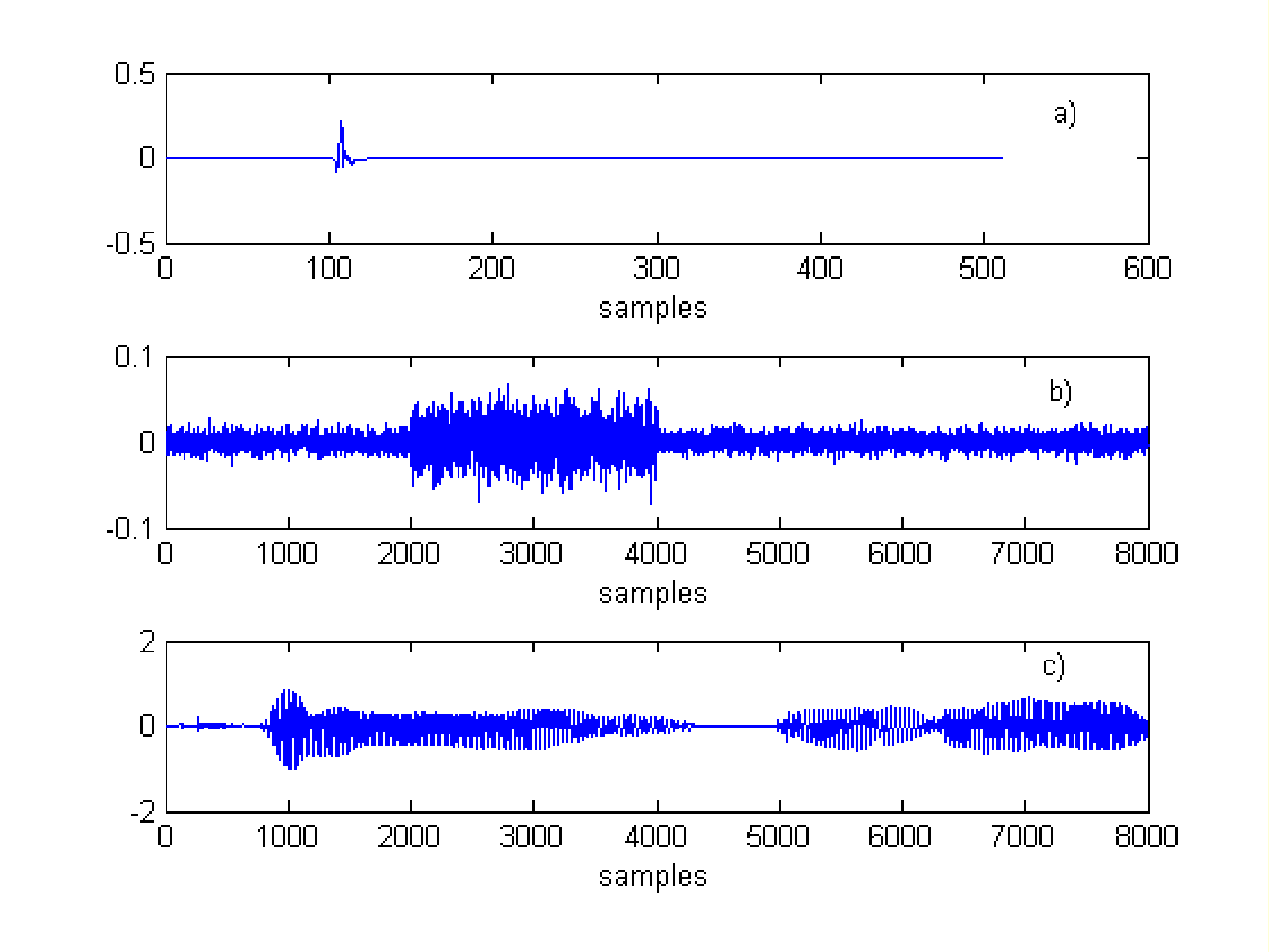
- *Numerical complexity in terms of multiplications for two situations: a) variable p, L=512; b) variable L, p=8*

$$IPAPA \Rightarrow L\left(p^2+3p+1\right)+O\left(p^3\right) \qquad FMIPAPA-DCD \Rightarrow 4L\left(p+1\right)+p^2+2$$



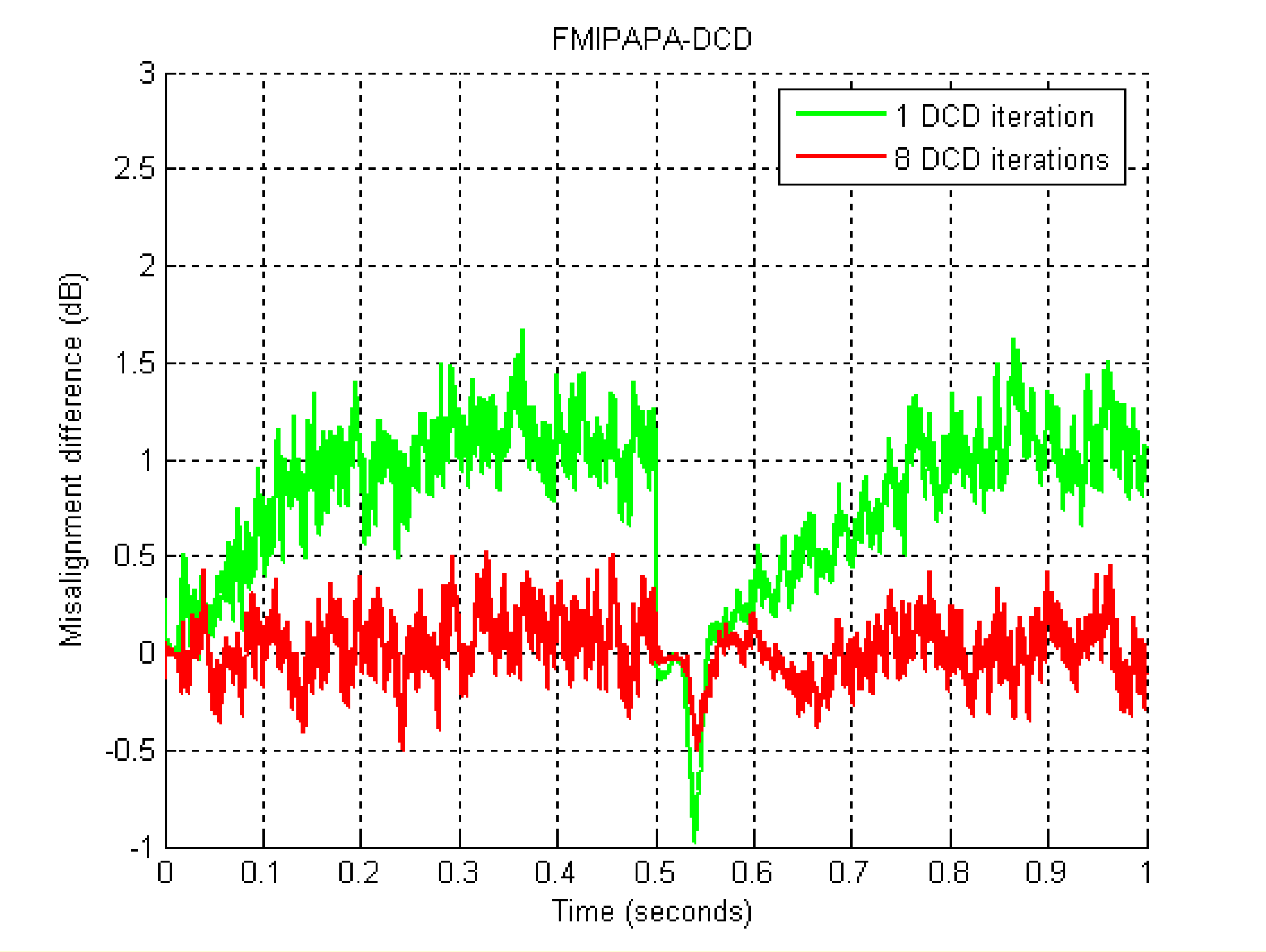
Simulation results

- *a) The echo path; b) the variable background noise; c) the speech signal*



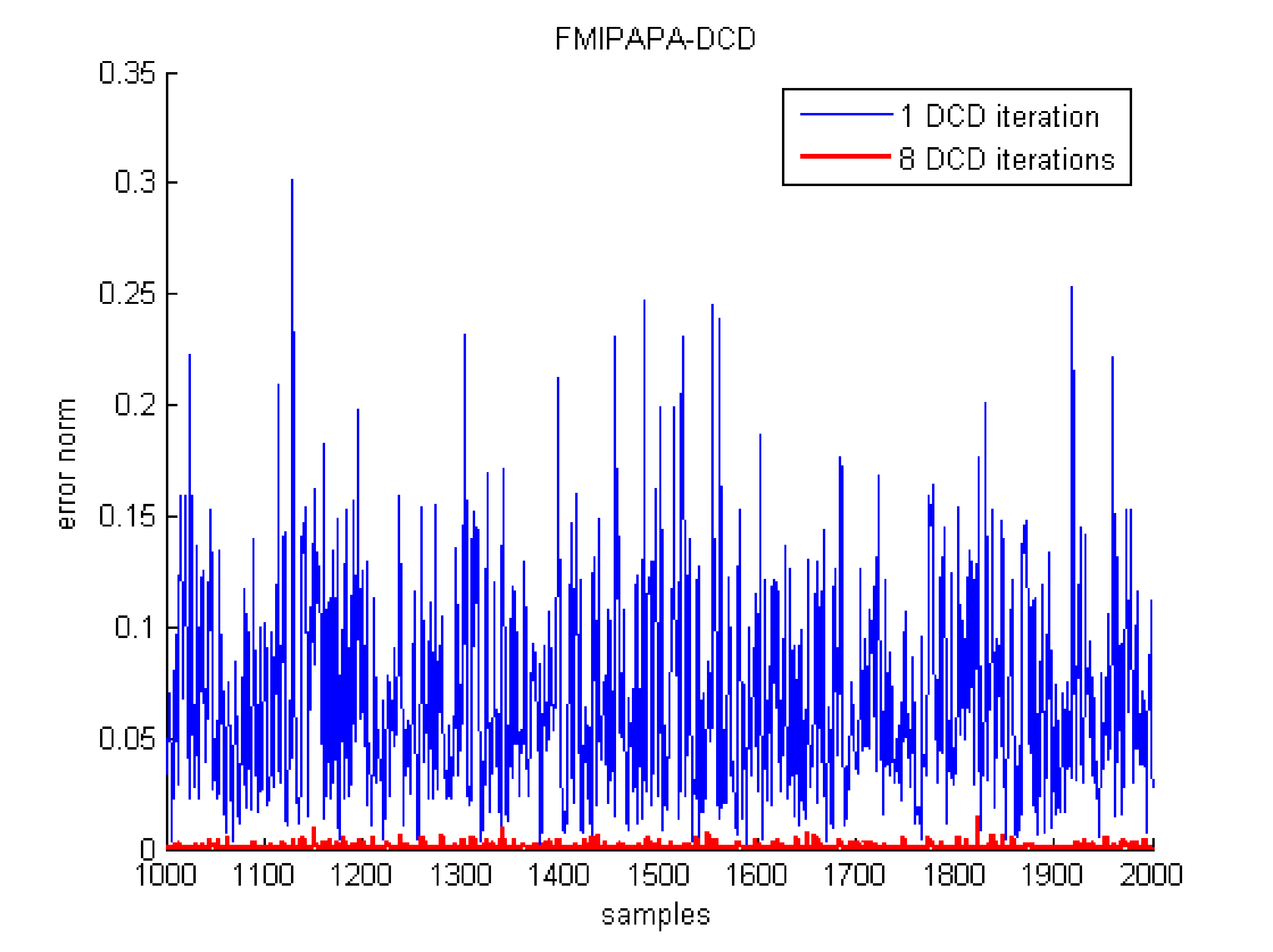
Simulation results

- *Misalignment difference between MIPAPA and FMIPAPA-DCD with different number of DCD iterations (1 and 8 respectively).*



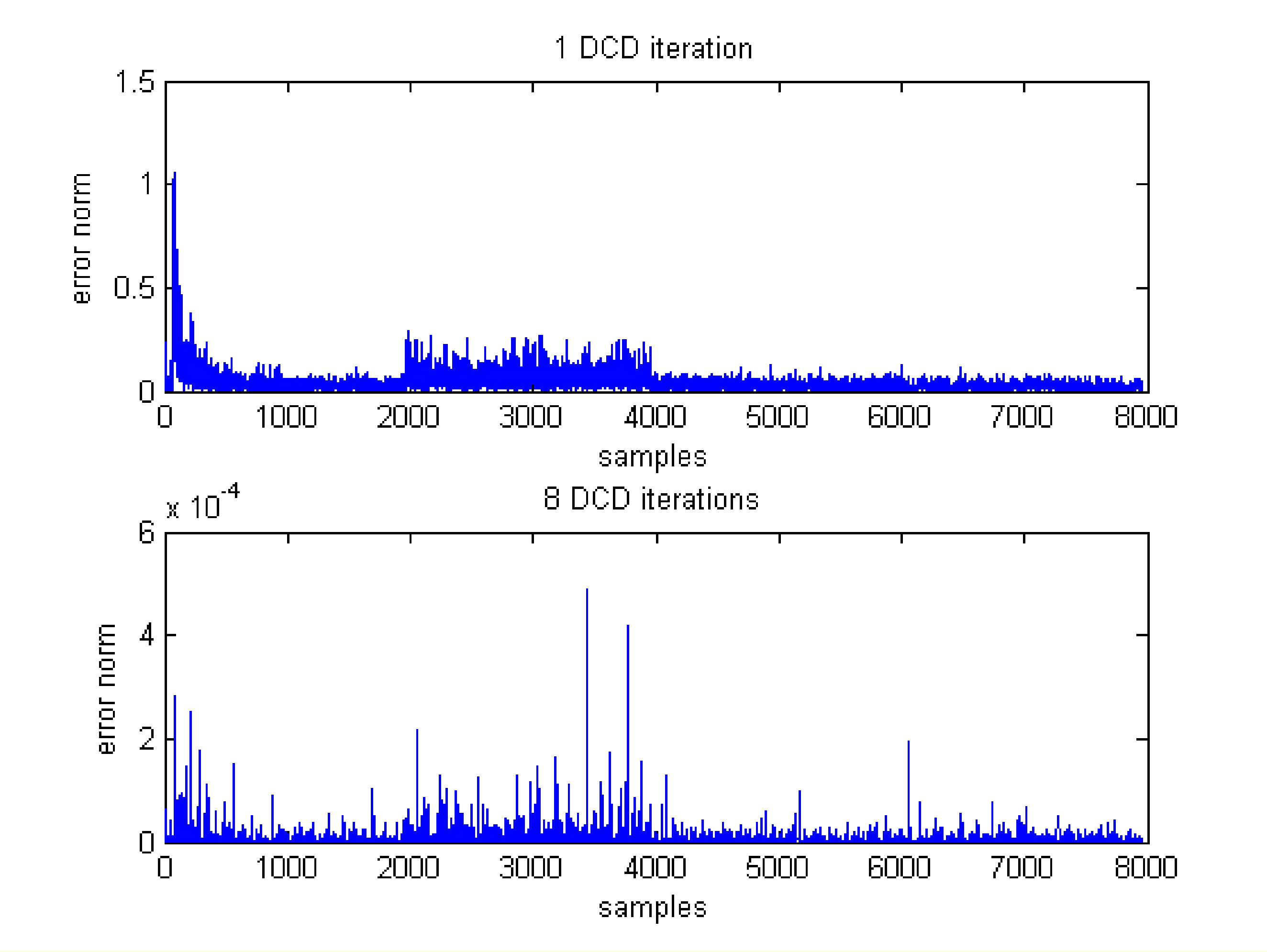
Simulation results

- ***The Error Norm for different number of DCD iterations for FMIPAPA-DCD. The input signal is a white Gaussian noise.***



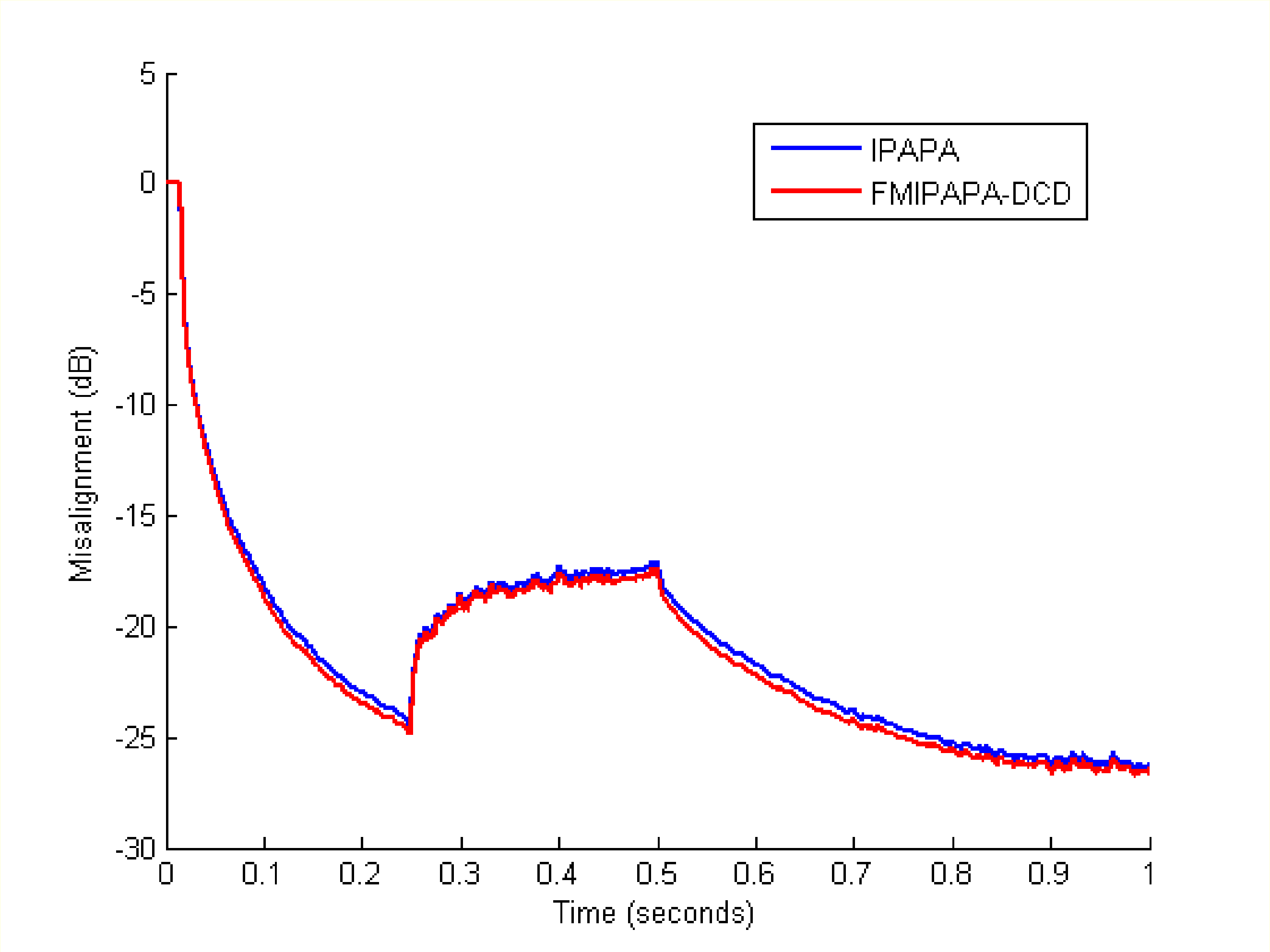
Simulation results

- *The Error Norm for different number of DCD iterations of FMIPAPA-DCD in case of variable background noise (SNR decreases from 20 dB to 10 dB between samples 2000 and 4000).*



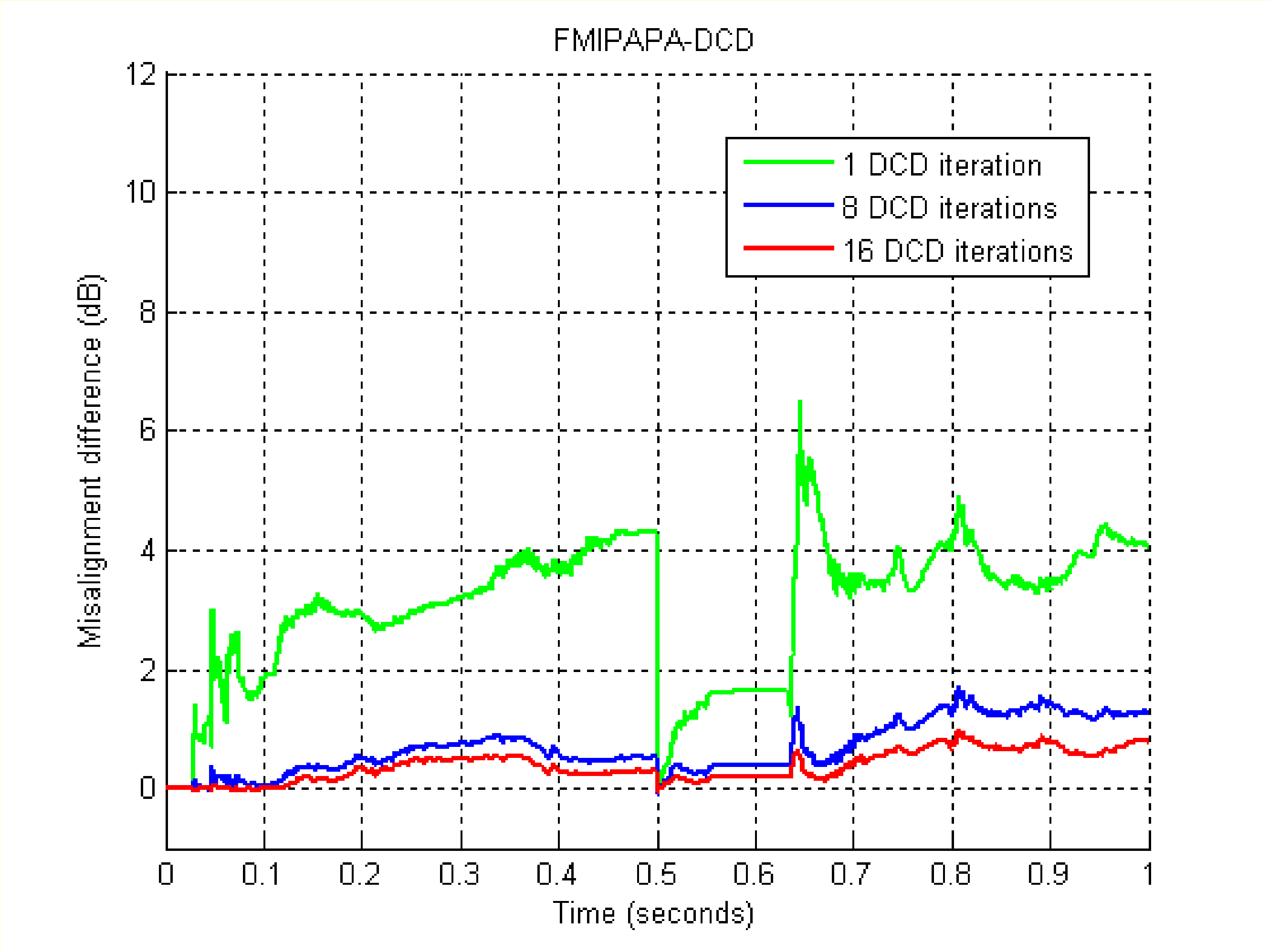
Simulation results

- *Misalignment of the IPAPA, and FMIPAPA-DCD. The input signal is a speech sequence, p = 8, L = 512, and variable background noise (SNR decreases from 30 dB to 10 dB between times 0.25 and 0.5, otherwise is 30 dB).*



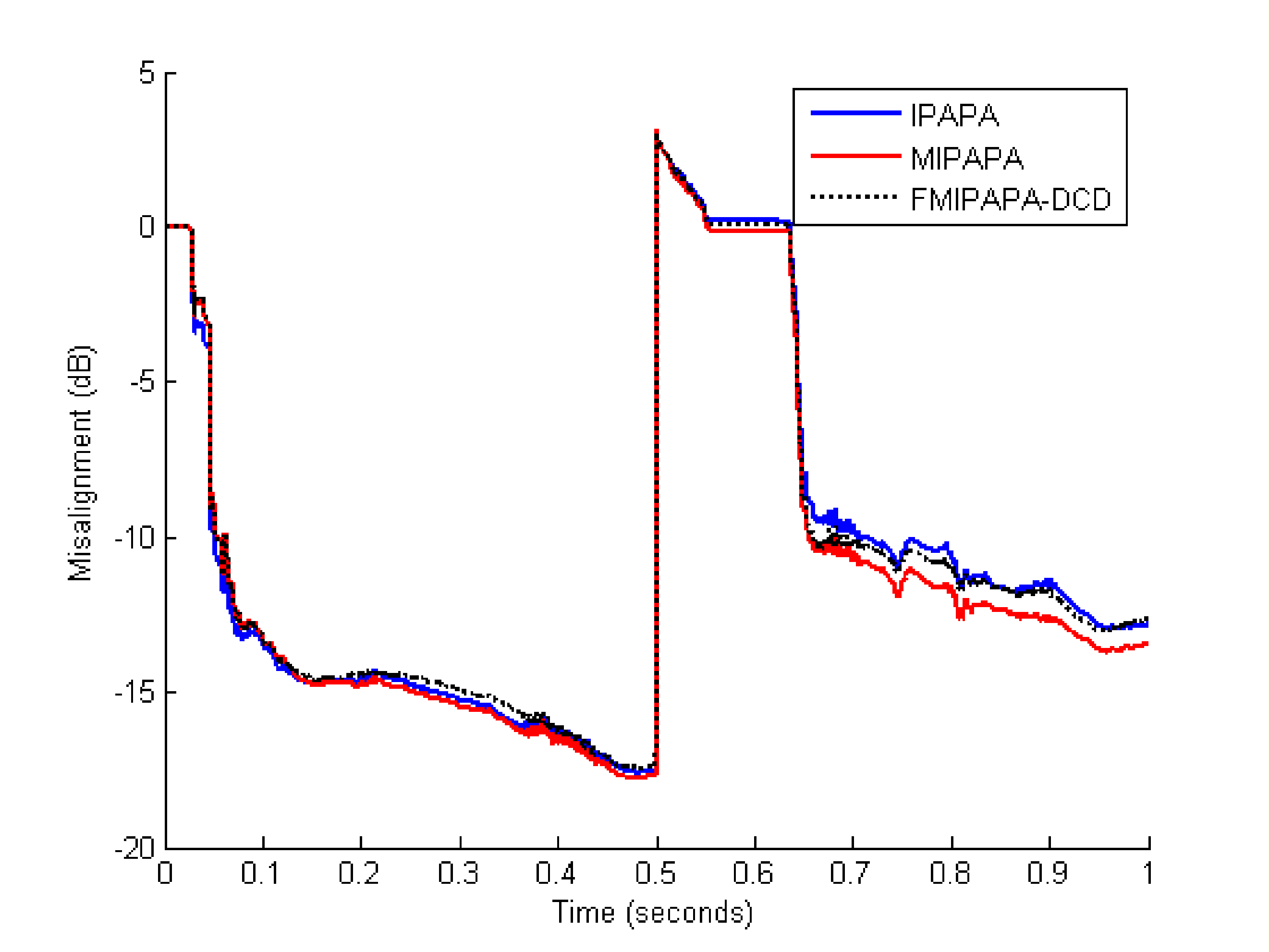
Simulation results

- *Misalignment difference between MIPAPA and FMIPAPA-DCD with different number of DCD iterations*



Simulation results

- *Misalignment of the IPAPA, MIPAPA, and FMIPAPA-DCD. The input signal is a speech sequence, p = 8, L = 512, SNR = 20 dB, echo path changes at time 0.5*



Conclusions

- FMIPAPA-DCD has been proposed for echo cancellation. It is improved version of IPAPA algorithm with reduced numerical complexity.
- A fast recursive filtering procedure is used. It exploits the time-shifting property of $\mathbf{P}'(n)$
- The influence of the number of DCD iterations on algorithm performance is investigated. As expected, if more DCD iterations are performed, better performances are obtained
- 8 DCD iterations only slightly alter the properties of the original algorithm

Relevant references

- D. L. Duttweiler, “Proportionate normalized least-mean-squares adaptation in echo cancellers,” *IEEE Trans. Speech Audio Process.*, vol. 8, no. 5, pp. 508–518, Sept. 2000.
- H. Deng and M. Doroslovački, “Proportionate adaptive algorithms for network echo cancellation,” *IEEE Trans. Signal Process.*, vol. 54, no. 5, pp. 1794–1803, May 2006.
- J. Benesty and S. L. Gay, “An improved PNLMS algorithm,” in *Proc. IEEE ICASSP*, 2002, pp. II-1881–II-1884.
- K. Ozeki and T. Umeda, “An adaptive filtering algorithm using an orthogonal projection to an affine subspace and its properties,” *Electron. Commun. Jpn.*, vol. 67-A, no. 5, pp. 19–27, May 1984.
- F. Albu, H.K. Kwan, “Fast block exact Gauss-Seidel pseudo affine projection algorithm”, *Electronics Letters*, Oct. 2004, pp. 1451-1453, Vol. 40, Issue:22
- Y. Zakharov and F. Albu, “Coordinate descent iterations in fast affine projection algorithm,” *IEEE Signal Processing Letters*, vol. 12, pp. 353–356, May 2005
- Y. Zakharov, “Low complexity implementation of the affine projection algorithm”, *IEEE Signal Processing Letters*, vol. 15, pp. 557-560, 2008
- F. Albu, C. Paleologu, J. Benesty, and S. Ciochina, “A low complexity proportionate affine projection algorithm for echo cancellation,” in *Proc. EUSIPCO*, Aalborg, Denmark, August 2010, pp. 6-10