Tensor-Based Models for Blind DS-CDMA Receivers

by Dimitri Nion and Lieven De Lathauwer

ETIS Lab., CNRS UMR 8051
6 avenue du Ponceau, 95014 CERGY
FRANCE

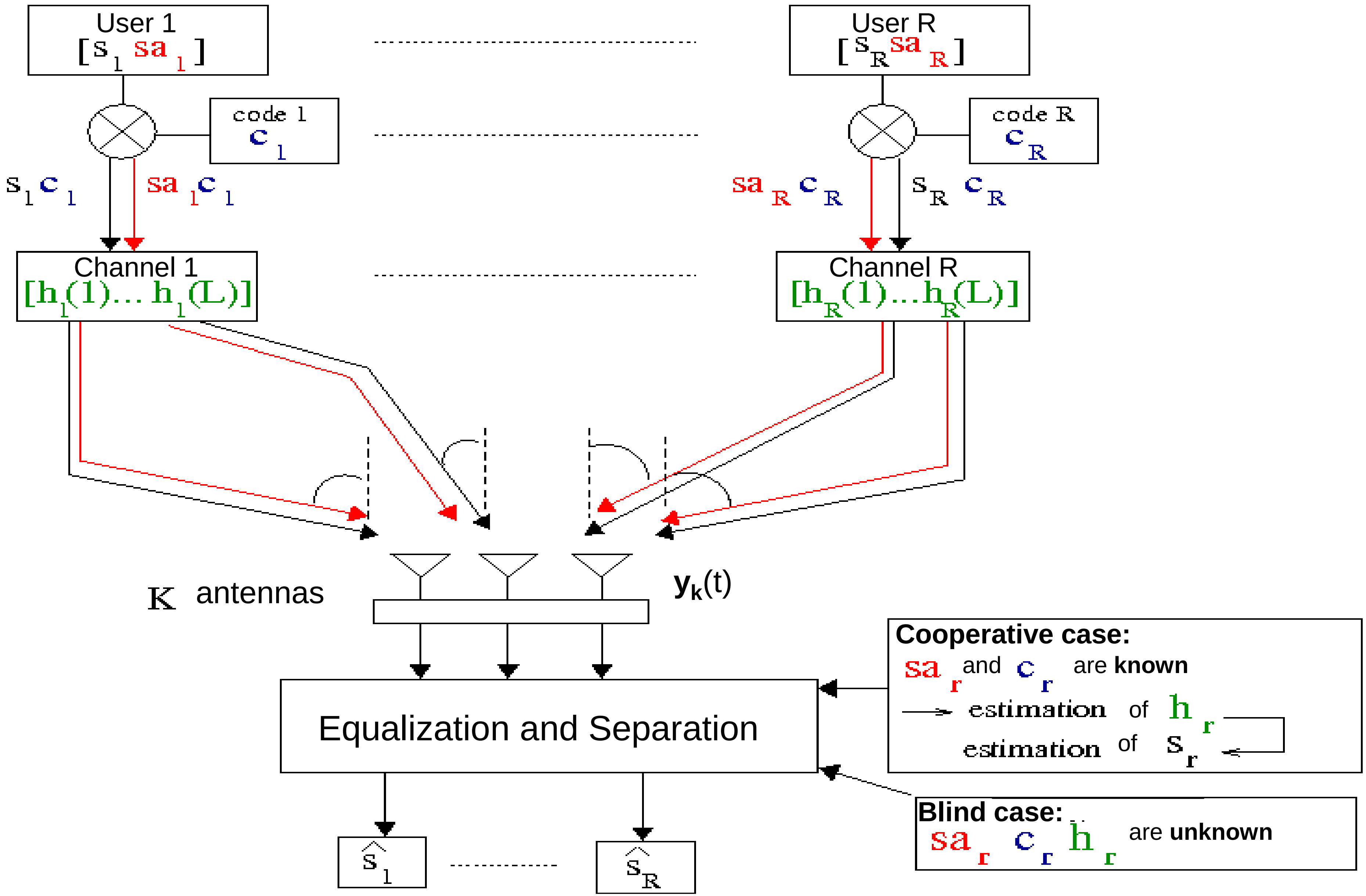
ASILOMAR 2007

November 4-7 2007, Pacific Grove, USA

Context

- Research Area: Blind Source Separation (BSS)
- Application: Wireless Communications (DS-CDMA system here)
- System: Multiuser DS-CDMA, uplink, antenna array receiver
- Propagation:
  - P1 Instantaneous channel (single path)
  - P2 Multipath Channel with Inter-Symbol-Interference (ISI) and far-field reflections only (from the receiver point of view)
  - P3 Multipath Channel (ISI) and reflections not only in the far-field (specular channel model)
- Assumptions: No knowledge of the channel, neither of CDMA codes, noise level and antenna array response (BLIND approach)
- Objective: Estimate each user's symbol sequence
- Method:
  - Deterministic: relies on multilinear algebra
  - How? store observations in a third order tensor and decompose it in a sum of users' contributions
- Idea: - Tensor Model « richer » than matrix model

DS-CDMA system: cooperative vs. blind



Blind Approach: Why?

Several motivations among others:

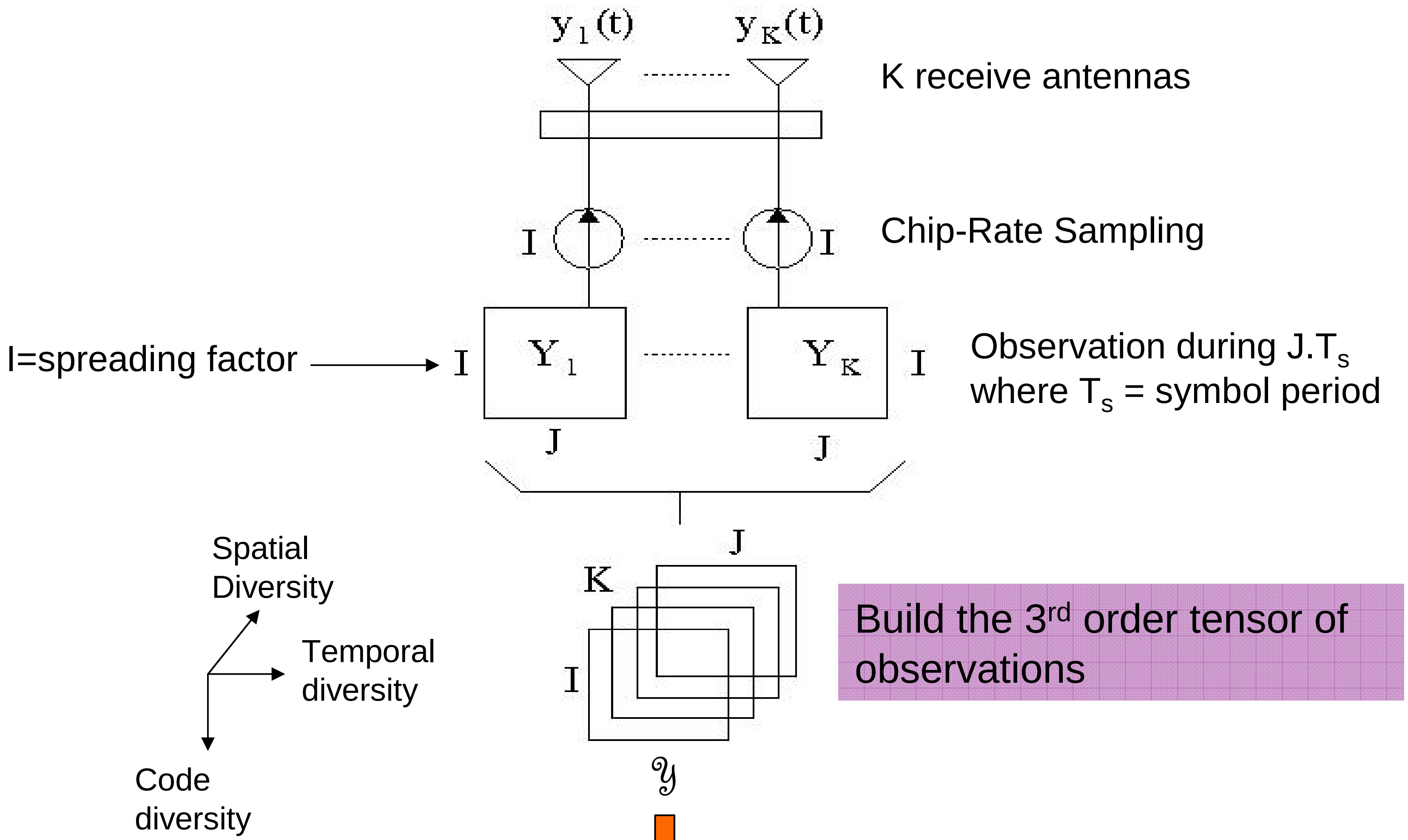
- Elimination or reduction of the learning frames: more than 40 % of the transmission rate devoted to training in UMTS

- Training not efficient in case of severe multipath fading or fast time varying channels

- Applications: eavesdropping, source localization, …

- If learning sequence unavailable or partially received

Blind Approach: How? (1)

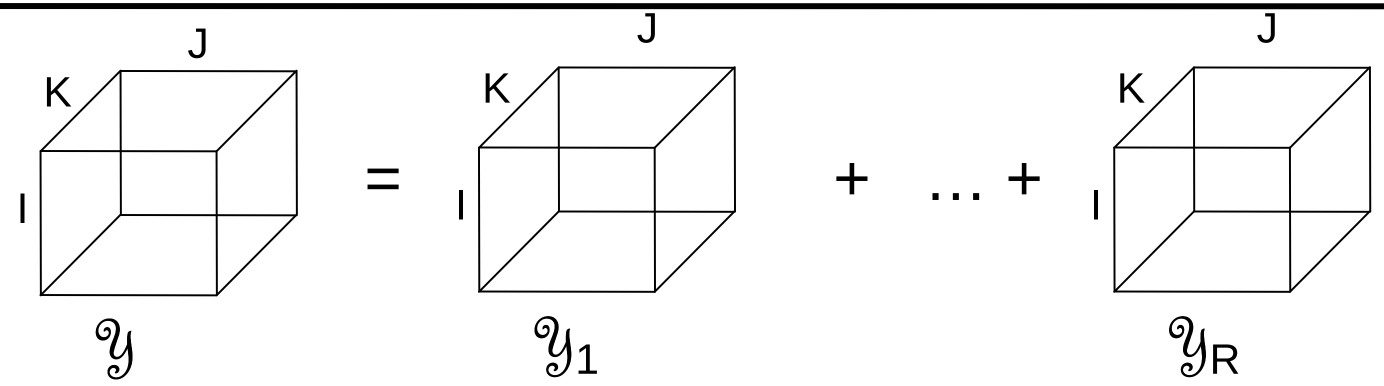


Build the 3rd order tensor of observations

Numerical processing:

Blind Equalization and Separation performed by decomposition of $\mathcal{Y}$

Blind Approach: How? (2)



Decomposition of $\mathcal{Y}$: sum of R users' contributions

Algebraic structure of $\mathcal{Y}_r$?	Estimation of $\mathcal{Y}_r$?	Identifiability of $\mathcal{Y}_r$?
↓ Different according to the propagation scenario	↓ Goal: Blind Separation and equalization	↓ Uniqueness of tensor decompositions
↓ Build different tensor decompositions	↓ Build algorithms to compute tensor decompositions	↓ Constraints on the number of users
Part I	Part II	Not in this talk

Introduction

I. Tensor Decompositions

1. Single path only (instantaneous channel):

→ PARAFAC decomposition

2. Multipath Channel with ISI and far-field reflections only :

→ Block-Component-Decomposition in rank-(L,L,1) terms : BCD(L,L,1)

3. Multipath Channel with ISI and reflections not only in the far-field:

→ Block-Component-Decomposition in rank-(L,P,.) terms : BCD(L,P,.)

II. Algorithms to compute tensor decompositions

II. Simulation Results

Conclusion and Perspectives

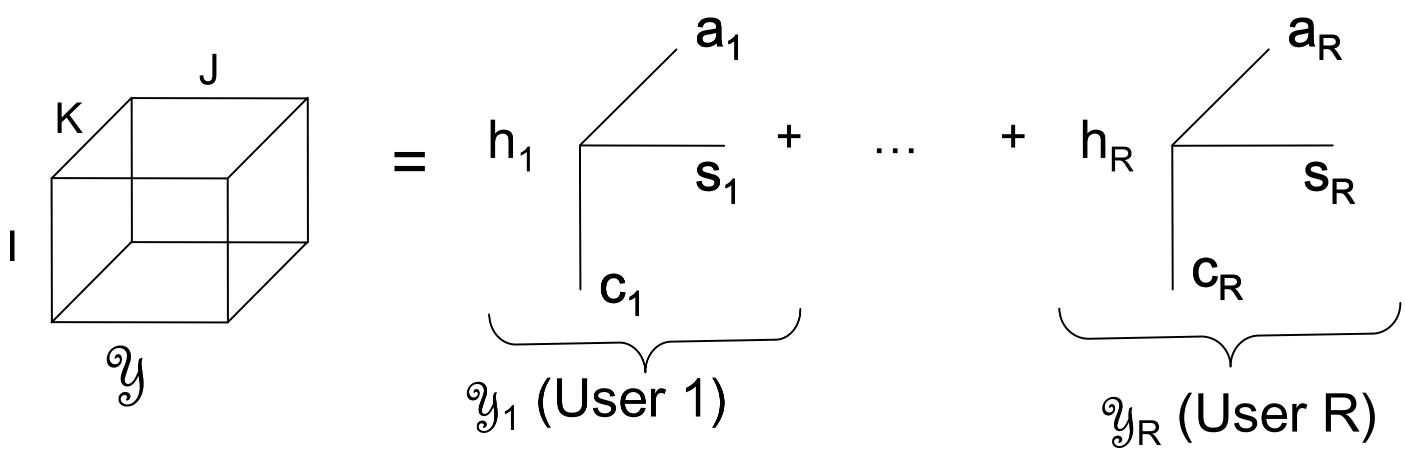
PARAFAC decomposition

If single path only (instantaneous mixture), $\mathcal{Y}$ follows a PARAFAC decomposition *[Sidiropoulos, Giannakis & Bro, 2000].*

Analytic Model:

$$y_{ijk} = \sum_{r=1}^{R} h_r c_{ir} s_{jr} a_{kr}$$

Algebraic Model:



c_r holds the I 'chips' r^{th} user's spreading code

a_r holds the response of the K antennas

s_r holds the J consecutive symbols transmitted by user r

h_r fading factor of the instantaneous channel

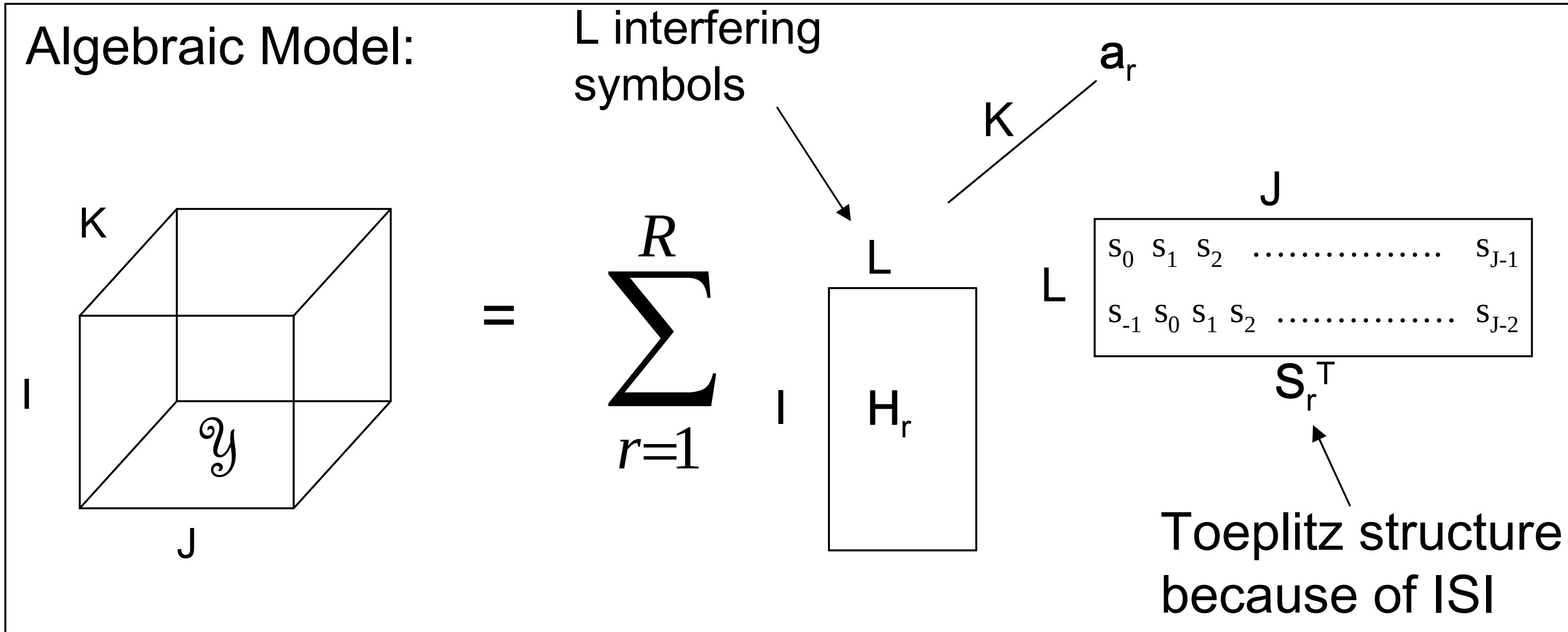
BCD-(L,L,1)

If multi-paths in the far field + ISI , $\mathcal{Y}$ follows a
« Block Component Decomposition in rank-(L,L,1) terms », BCD-(L,L,1)
[De Lathauwer & De Baynast, 2003], [Nion & De Lathauwer, SPAWC 2007].

Analytic Model:

$$y_{ijk} = \sum_{r=1}^{R} a_{kr} \sum_{l=1}^{L} h_r(i + (l-1)I)\, s^{(r)}_{j-l+1}$$

Algebraic Model:



BCD-(L,P,.)

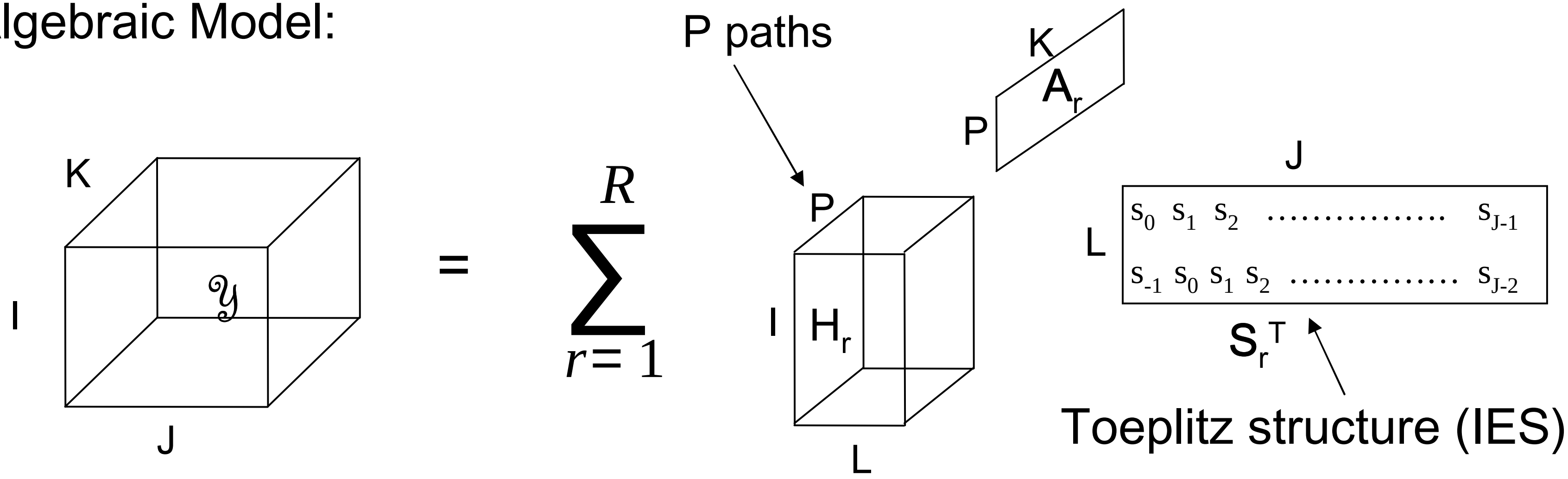
If multi-paths not only in the far-field + ISI , $\mathcal{Y}$ follows a BCD-(L,P,.)
[Nion & De Lathauwer, ICASSP 2005].

Analytic Model:

$$y_{ijk} = \sum_{r=1}^{R} \sum_{p=1}^{P} a_k(\theta_{rp}) \sum_{l=1}^{L} h_{rp}(i + (l-1)I) s_{j-l+1}^{(r)}$$

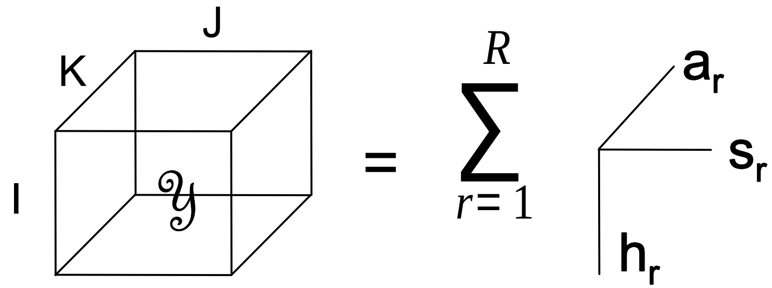
1 path = 1 delay, 1angle of arrival and 1 fading coefficient

Algebraic Model:



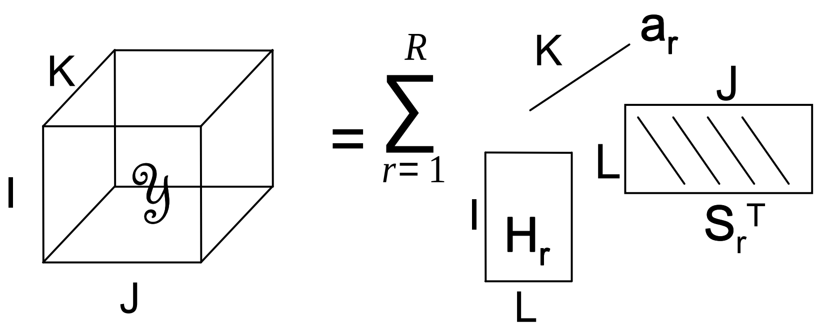
Unknowns for each decomposition

PARAFAC



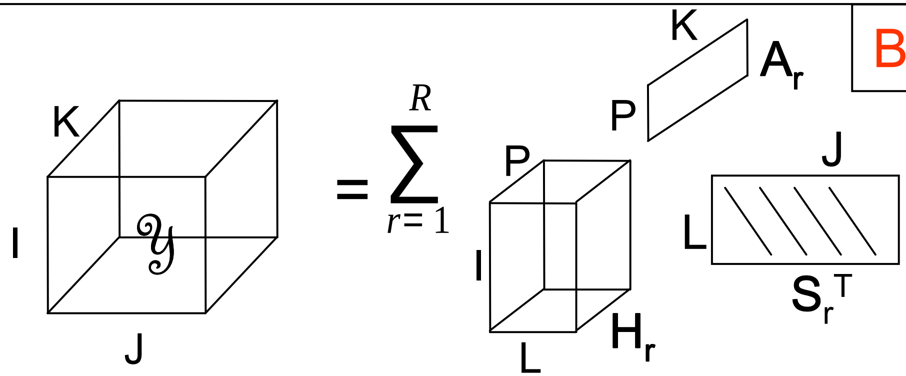
$$\left\{\begin{array}{l}\mathbf{H} \in C^{I \times R} \\ \mathbf{S} \in C^{J \times R} \\ \mathbf{A} \in C^{K \times R}\end{array}\right.$$

BCD-(L,L,1)



$$\left\{\begin{array}{l}\mathbf{H} \in C^{I \times RL} \\ \mathbf{S} \in C^{J \times RL} \text{ Block-Toeplitz} \\ \mathbf{A} \in C^{K \times R}\end{array}\right.$$

BCD-(L,P,.)



$$\left\{\begin{array}{l}\mathbf{H} \in C^{I \times RPL} \\ \mathbf{S} \in C^{J \times RL} \text{ Block-Toeplitz} \\ \mathbf{A} \in C^{K \times RP}\end{array}\right.$$

Introduction

I. Les décompositions tensorielles

II. Algorithms to compute Tensor Decompositions

1. Algorithm 1: ALS (“Alternating Least Squares”)

2. Algorithm 2: ALS + LS (“Line Search”)

3. Algorithm 3: LM (“Levenberg-Marquardt”)

III. Simulation Results

Conclusion et Perspectives

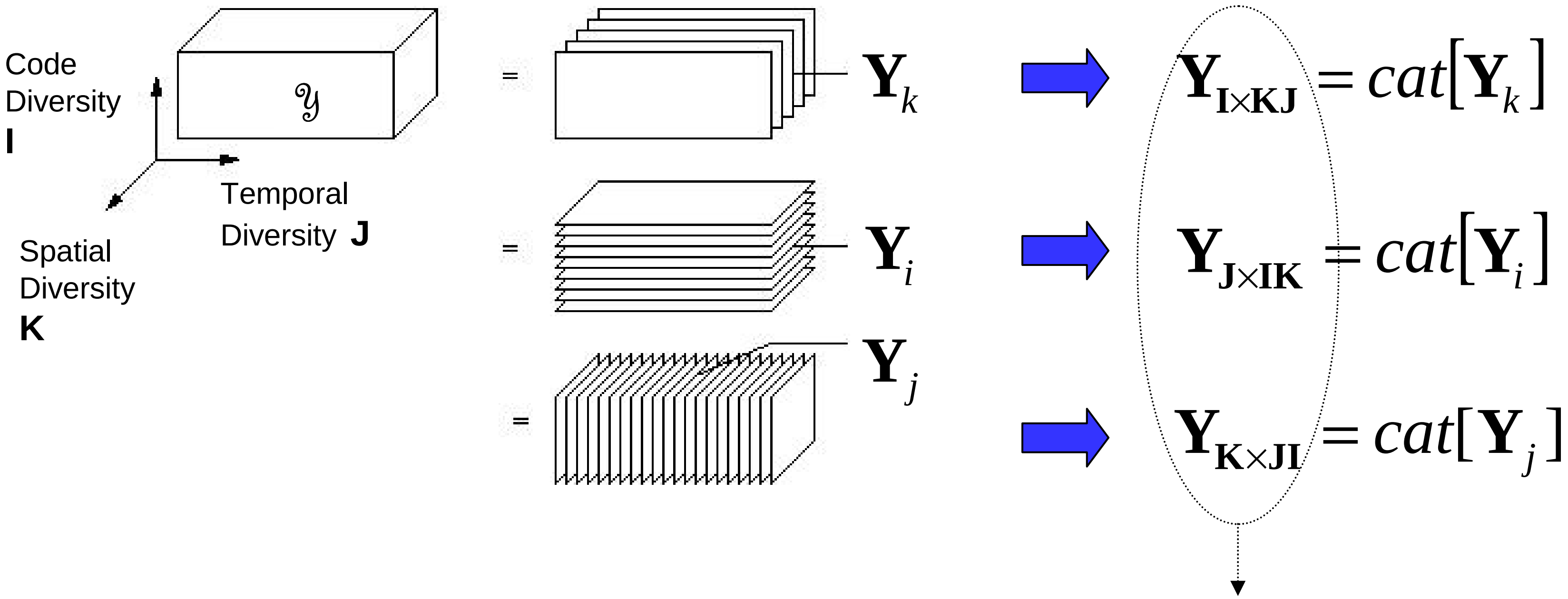
Objective of the proposed algorithms

- Decomposition of $\mathcal{Y}$ ⟺ Estimation of components **A**, **S** and **H**
- Minimize frobenius norm of residuals. Cost function:

$$\Phi = \left\| \mathcal{Y} - Tens(\hat{\mathbf{H}}, \hat{\mathbf{S}}, \hat{\mathbf{A}}) \right\|_F^2$$

Tens = PARAFAC or DCB-(L,L,1) or DCB-(L,P,.)

Useful Tool: « Matricize » the tensor of observations



3 matrix representations of the same tensor

Algorithm 1: ALS « Alternating Least Squares »

➢ Principle: Alternate between least squares update of the 3 matrices $A=[A_1,\ldots,A_R]$, $S=[S_1,\ldots,S_R]$ et $H=[H_1,\ldots,H_R]$.

$$\text{Initialization}: \hat{\mathbf{A}}^{(0)}, \hat{\mathbf{H}}^{(0)}, k=1$$

$$\textit{while} \left|\Phi^{(k-1)} - \Phi^{(k)}\right| > \varepsilon \quad (\text{e.g. } \varepsilon = 10^{-6})$$

$$\hat{\mathbf{S}}^{(k)} = \mathbf{Y}_{\mathbf{J}\times\mathbf{IK}} \cdot \left[\mathbf{Z}_1(\hat{\mathbf{A}}^{(k-1)}, \hat{\mathbf{H}}^{(k-1)})\right]^{\dagger} \qquad (1)$$

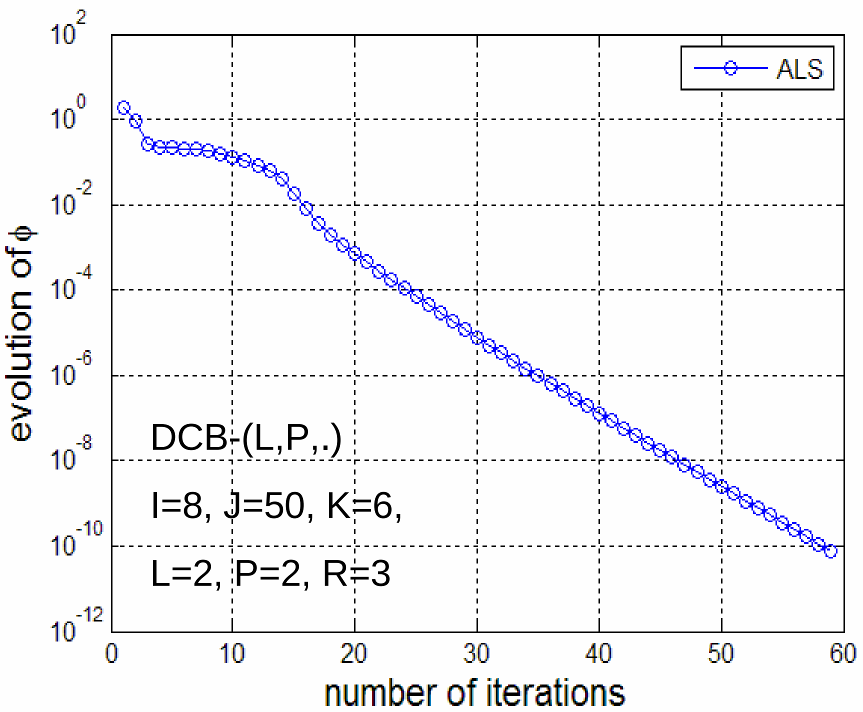
$$\hat{\mathbf{H}}^{(k)} = \mathbf{Y}_{\mathbf{I}\times\mathbf{KJ}} \cdot \left[\mathbf{Z}_2(\hat{\mathbf{S}}^{(k)}, \hat{\mathbf{A}}^{(k-1)})\right]^{\dagger} \qquad (2)$$

$$\hat{\mathbf{A}}^{(k)} = \mathbf{Y}_{\mathbf{K}\times\mathbf{JI}} \cdot \left[\mathbf{Z}_3(\hat{\mathbf{H}}^{(k)}, \hat{\mathbf{S}}^{(k)})\right]^{\dagger} \qquad (3)$$

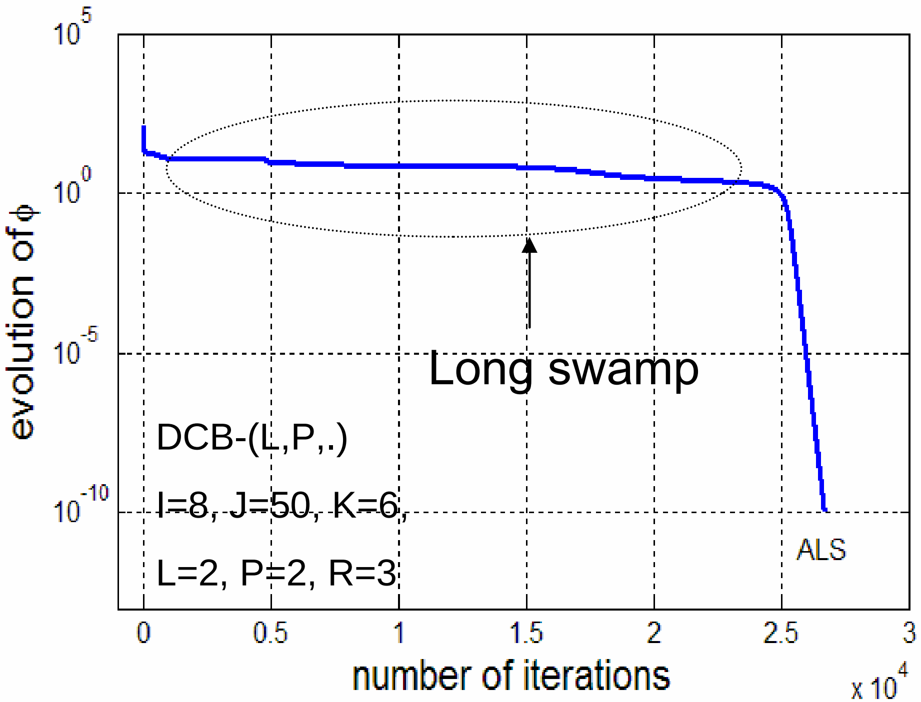
$$k \leftarrow k+1$$

Convergence of ALS

« Easy » Problem



«Difficult» Problem



Because of long swamps that might occur, we propose 2 algorithms that improve convergence speed.

Algorithm 2: Insert a Line Search step in ALS

For each iteration, perform linear interpolation of the 3 components A, H and S from their values at the 2 previous iterations.

Iteration k

1.Line Search:

$$\hat{\mathbf{S}}^{(new)} = \hat{\mathbf{S}}^{(k-2)} + \rho\,(\hat{\mathbf{S}}^{(k-1)} - \hat{\mathbf{S}}^{(k-2)})$$

$$\hat{\mathbf{A}}^{(new)} = \hat{\mathbf{A}}^{(k-2)} + \rho\,(\hat{\mathbf{A}}^{(k-1)} - \hat{\mathbf{A}}^{(k-2)})$$

$$\hat{\mathbf{H}}^{(new)} = \hat{\mathbf{H}}^{(k-2)} + \rho\,(\hat{\mathbf{H}}^{(k-1)} - \hat{\mathbf{H}}^{(k-2)})$$

Directions of research

Choice of step ρ important

Can be optimally calculated with « Enhanced Line Search with Complex Step» (ELSCS)

2. ALS update

$$\hat{\mathbf{S}}^{(k)} = \left[\tilde{\mathbf{Z}}_1(\hat{\mathbf{A}}^{(new)}, \hat{\mathbf{H}}^{(new)})\right]^{\dagger} \cdot Y_{JIK}$$

$$\hat{\mathbf{H}}^{(k)} = \mathbf{Y}_{\mathbf{I}\times\mathbf{KJ}} \cdot \left[\mathbf{Z}_2(\hat{\mathbf{S}}^{(k)}, \hat{\mathbf{A}}^{(new)})\right]^{\dagger}$$

$$\hat{\mathbf{A}}^{(k)} = \mathbf{Y}_{\mathbf{K}\times\mathbf{JI}} \cdot \left[\mathbf{Z}_3(\hat{\mathbf{H}}^{(k)}, \hat{\mathbf{S}}^{(k)})\right]^{\dagger}$$

$$k \leftarrow k+1$$

Algorithm 3: LM « Levenberg-Marquardt »

- Concatenate vectorized unknowns vec(A), vec(H) and **s** in a long vector **p**
- Update p: $\mathbf{p}^{(k+1)} = \mathbf{p}^{(k)} + \Delta\mathbf{p}^{(k)}$ (1)
- Gauss-Newton: $(\mathbf{J}^H\mathbf{J})\Delta\mathbf{p}^{(k)} = -\mathbf{g}$ (2)
- Levenberg-Marquardt: $(\mathbf{J}^H\mathbf{J} + \lambda\mathbf{I})\Delta\mathbf{p}^{(k)} = -\mathbf{g}$ (3)

- The matrix $(\mathbf{J}^H\mathbf{J} + \lambda\mathbf{I})$ is positive definite: solve (3) by Cholesky decomposition and Gaussian elimination.

- According to the condition number of $\mathbf{J}^H\mathbf{J} + \lambda\,\mathbf{I}$, update λ in each iteration.
  - If $(\mathbf{J}^H\mathbf{J} + \lambda\mathbf{I})$ ill-conditioned then increase λ :

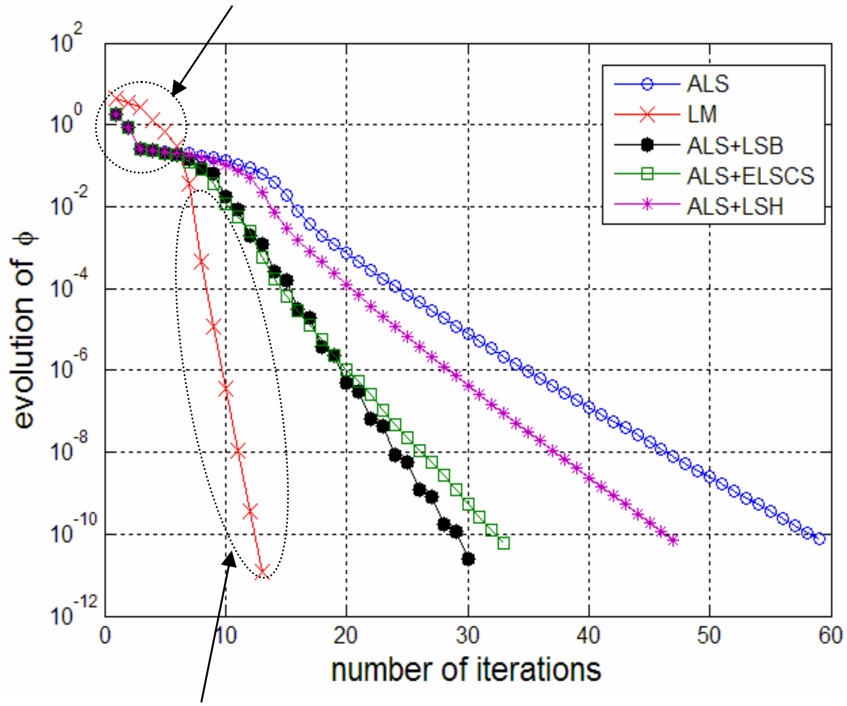
    get closer to gradient descent update $\Delta\mathbf{p}^{(k)} \approx -\frac{1}{\lambda}\mathbf{g}$
  - If $(\mathbf{J}^H\mathbf{J} + \lambda\mathbf{I})$ well-conditioned then decrease λ:

    get closer to Gauss-Newton update $(\mathbf{J}^H\mathbf{J})\Delta\mathbf{p}^{(k)} \approx -\mathbf{g}$

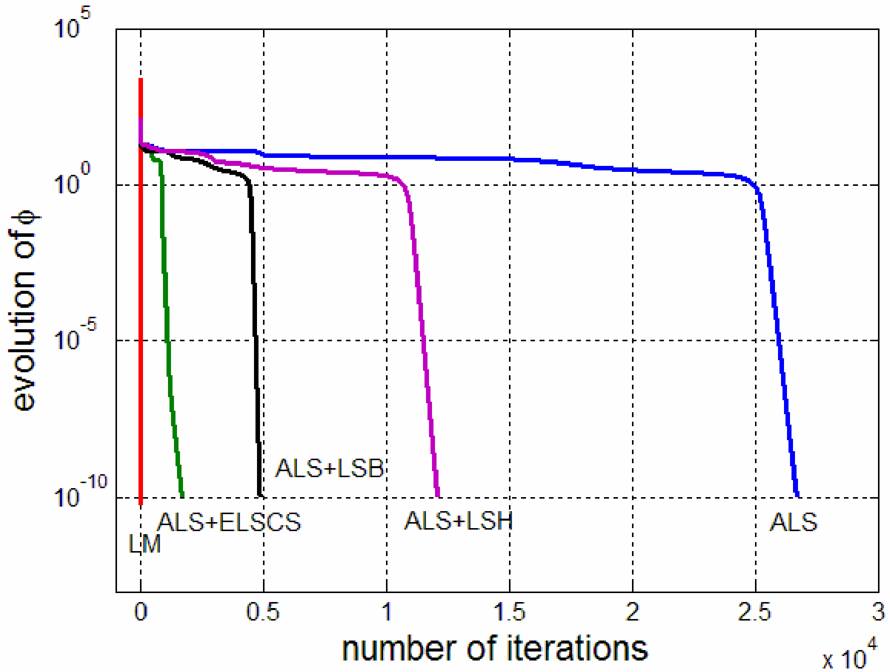
Convergence of algorithms ALS, ALS+LS et LM

«easy» problem	«difficult» problem

Gradient Descent



Gauss Newton (quadratic convergence)



LM and ALS+ELSCS converge much faster than standard ALS, especially for difficult problems: the length of swamps is considerably reduced.

Introduction

I. Tensor Decompositions

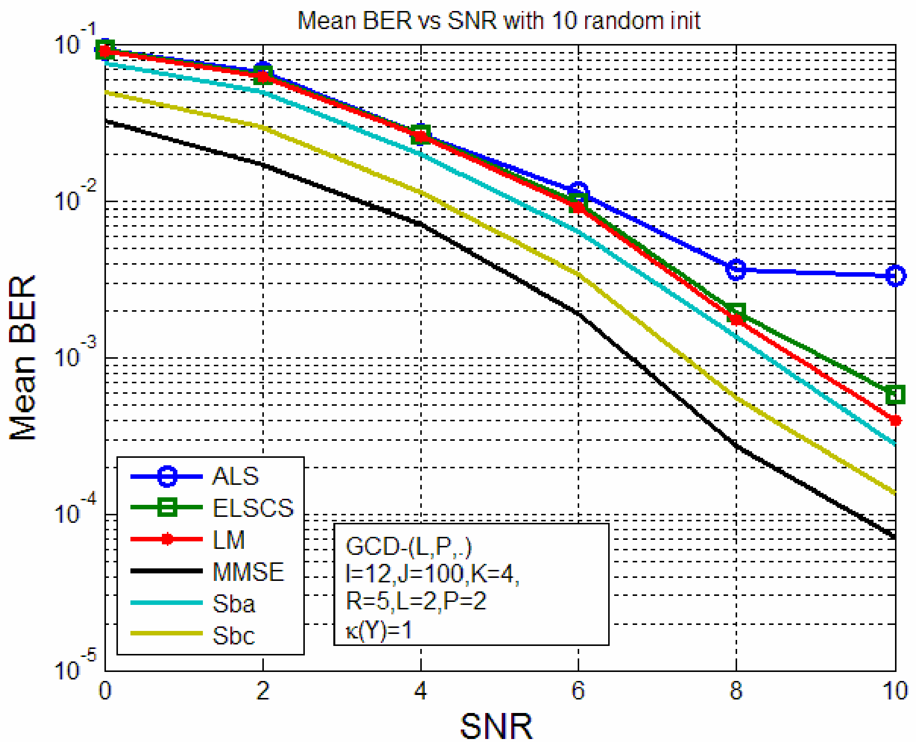
II. Algorithms to compute Tensor Decompositions

III. Simulation Results

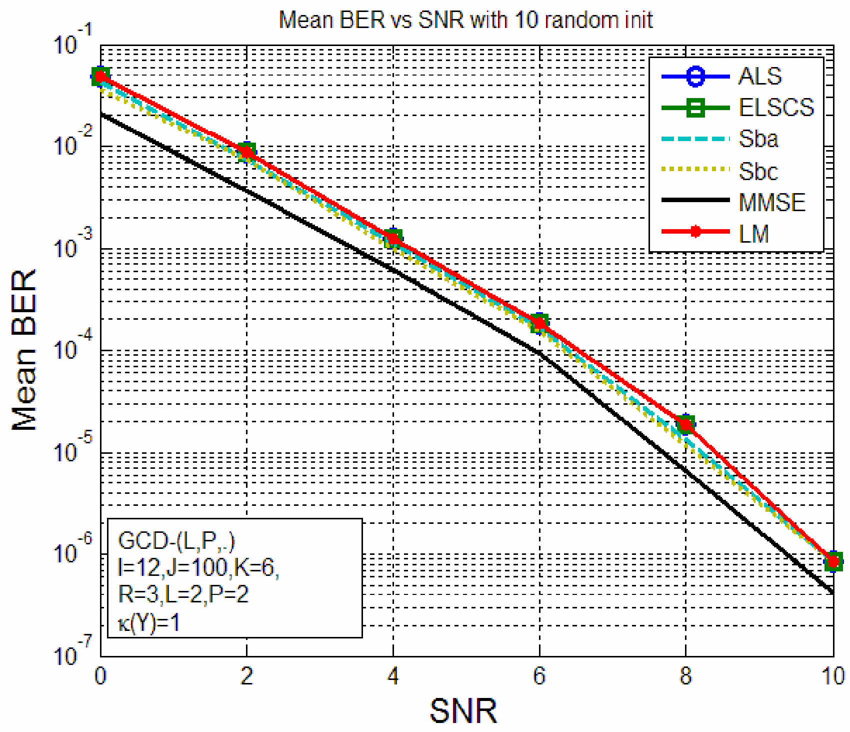
Conclusion et Perspectives

Impact of number of antennas

BCD-(L,P,.) with: spreading factor I=12, J=100 symbols, L=2 interfering symbols, P=2 paths per user and 10 random initializations, + AWGN



K=4 antennas and R=5 users



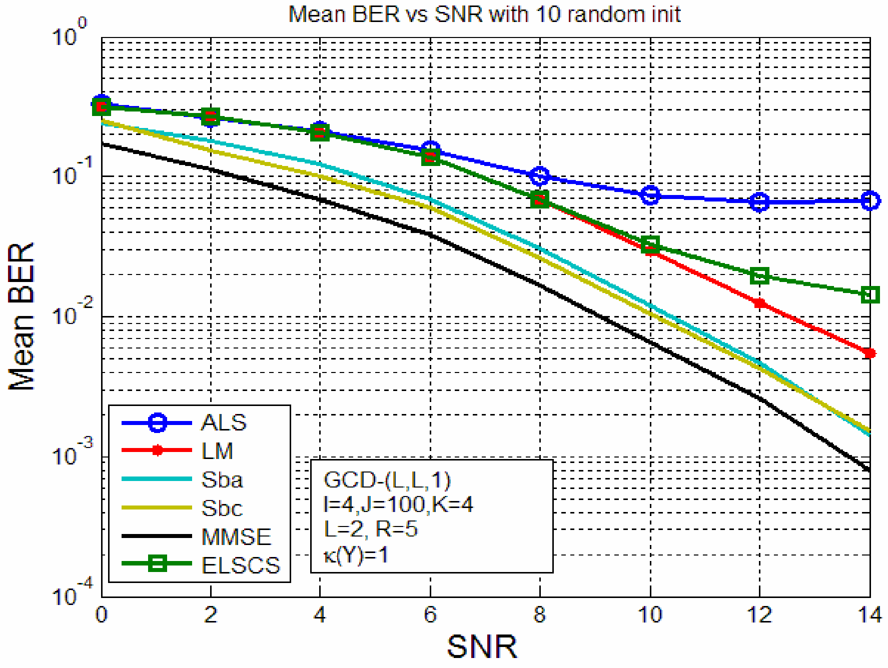
K=6 antennas and R=3 users

Impact of Near-Far effect

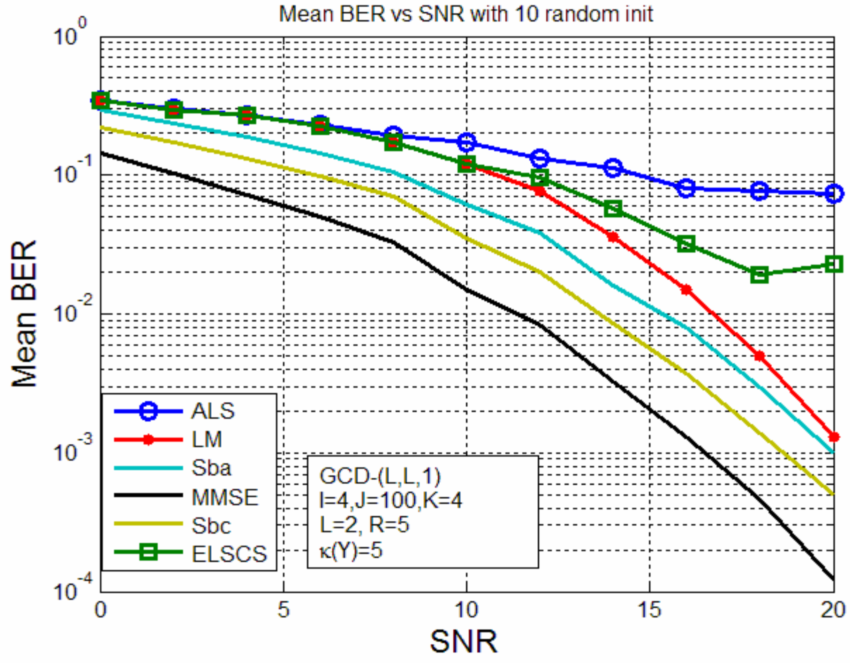
$$\mathcal{Y} = \sum_{r=1}^{R} \alpha_r \frac{\mathcal{Y}_r}{\left\| \mathcal{Y}_r \right\|_F} + \mathcal{B} \qquad \kappa(\mathcal{Y}) = \frac{\max(\alpha_r)}{\min(\alpha_r)}$$

BCD-(L,L,1) with spreading factor I=4, J=100 symbols, K=4 antennas, L=2 interfering symbols, R=5 users and 10 random initializations, + AWGN

Note: more users than antennas (R>K) and overloaded system (R>I)



$\kappa(\mathcal{Y}) = 1$



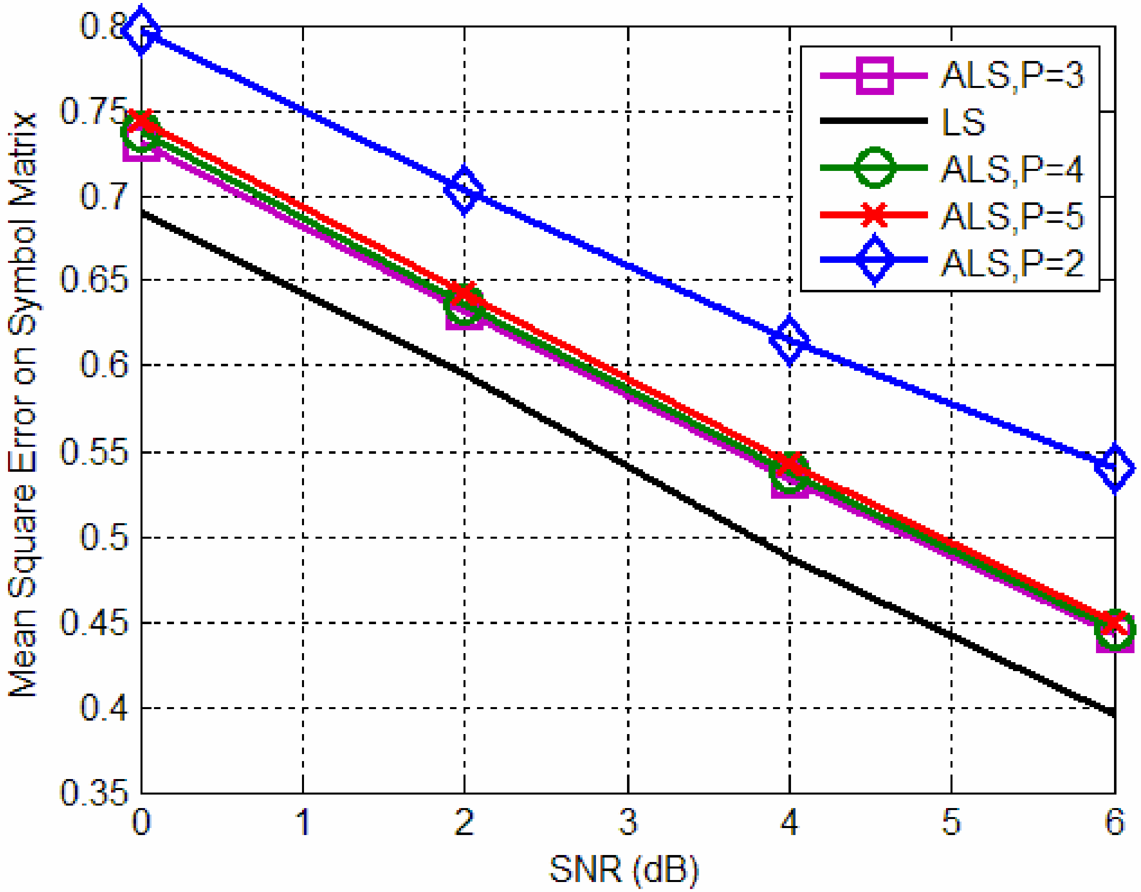
$\kappa(\mathcal{Y}) = 5$

Over-estimation of the number of paths P

$\mathcal{Y}$ built with P=3 paths for each user.

Decomposition calculated with over-estimation of P (P=4 and P=5) and under-estimation of P (P=2).

MSE of symbol matrix vs. SNR



Conclusion

Tensor Models:

- PARAFAC receiver: ok if single path (instantaneous mixture)
- BCD receivers: multipaths + ISI (blind separation and equalization)

Approach:

- Deterministic, exploits multi-linearity of received signal, i.e. algebraic structure of tensor of observations. 1 diversity = 1 dimension of this tensor.

Algorithms:

- standard ALS sensitive to swamps that appear with ill-conditioned data or severe Near-Far effect
- ALS+ELSCS and LM offers much better performance.

Performances:

- Blind BCD receivers potentially very close to MMSE, provided that enough diversity is exploitable.

Uniqueness (not in this talk):

- Maximum number of users admissible in the system depends on the dimensions of the problem.