Blind Linear and Nonlinear Mixture Identification Using Source Sparsity Assumptions

Matthieu Puigt

LISIC - ULCO
IUT de Saint-Omer Dunkerque
matthieu.puigt@lisic.univ-littoral.fr
http://www-lisic.univ-littoral.fr/~puigt

Work done at FORTH-ICS, Heraklion, Crete, Greece

December 20, 2012

FORTH
Institute of Computer Science

FOundation for Research and Technology – Hellas (FORTH)

- FORTH
  - Public institution (1983) funded by projects
  - Largest Greek research centre, strong participation in EU projects
  - Seven research institutes in four different locations, one start-up incubator, one technology transfer organization, the Crete University press, the Observatory of Skinakas
- Institute of Computer Science (FORTH-ICS)
  - since 1983, internationally recognised, ERCIM member, 9 laboratories
- Signal Processing Lab (SPL)
  - 3 permanent researchers, 7 assistant researchers and 6 post-docs
  - Since 2005: 8 European Commission, 6 national, 3 industrial projects: more than 5.1 M€ in actual funding for FORTH. 6 active projects
  - Research interest: Statistical signal processing with applications in wireless sensor networks, image/video, audio and speech processing.

Outline of the talk

Part I

Basic introduction to blind source separation

Part II

Post-nonlinear sparse component analysis

Part III

Real-time source localization: a brief introduction

Let's talk about linear systems

All of you know how to solve this kind of systems:

$$\left\{ \begin{array}{rcl} 2 \cdot s_1 + 3 \cdot s_2 & = & 5 \\ 3 \cdot s_1 - 2 \cdot s_2 & = & 1 \end{array} \right. \quad (1)$$

If we resp. define A, $\underline{s}$, and $\underline{x}$ the matrix and the vectors:

$$A = \begin{bmatrix} 2 & 3 \\ 3 & -2 \end{bmatrix}, \underline{s} = [s_1, s_2]^T, \text{ and } \underline{x} = [5, 1]^T$$

Eq. (1) begins

$$\underline{x} = A \cdot \underline{s}$$

and the solution reads:

$$\underline{s} = A^{-1} \cdot \underline{x} = [1, 1]^T$$

Let's talk about linear systems

All of you know how to solve this kind of systems:

$$\left\{ \begin{array}{rcl} ? \cdot s_1 + ? \cdot s_2 & = & 5 \\ ? \cdot s_1 + ? \cdot s_2 & = & 1 \end{array} \right. \quad (1)$$

If we resp. define A, $\underline{s}$, and $\underline{x}$ the matrix and the vectors:

$$A = \begin{bmatrix} ? & ? \\ ? & ? \end{bmatrix}, \underline{s} = [s_1, s_2]^T, \text{ and } \underline{x} = [5, 1]^T$$

Eq. (1) begins

$$\underline{x} = A \cdot \underline{s}$$

and the solution reads:

$$\underline{s} = A^{-1} \cdot \underline{x} = ?$$

Let's talk about linear systems

All of you know how to solve this kind of systems:

$$\begin{cases} a_{11} \cdot s_1 + a_{12} \cdot s_2 & = & 5 \\ a_{21} \cdot s_1 + a_{22} \cdot s_2 & = & 1 \end{cases} \quad (1)$$

If we resp. define A, $\underline{s}$, and $\underline{x}$ the matrix and the vectors:

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}, \underline{s} = [s_1, s_2]^T, \text{ and } \underline{x} = [5, 1]^T$$

Eq. (1) begins

$$\underline{x} = A \cdot \underline{s}$$

and the solution reads:

$$\underline{s} = A^{-1} \cdot \underline{x} = \textbf{?}$$

How can we solve this kind of problem???

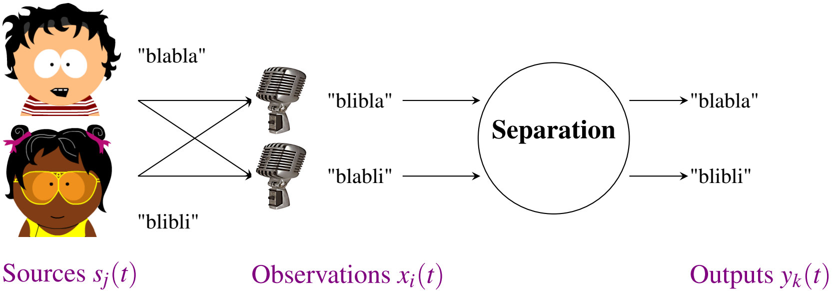
This problem is called **Blind Source Separation**.

Blind Source Separation problem

- N unknown sources s_j.
- One unknown operator $\mathcal{A}$.
- P observed signals x_i with the global relation

$$\underline{x} = \mathcal{A}\left(\underline{s}\right).$$

Goal: Estimating the vector $\underline{s}$, up to some indeterminacies.

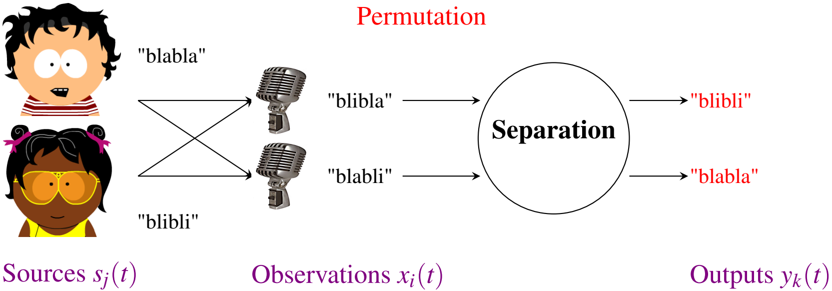


Blind Source Separation problem

- N unknown sources s_j.
- One unknown operator $\mathcal{A}$.
- P observed signals x_i with the global relation

$$\underline{x} = \mathcal{A}\left(\underline{s}\right).$$

Goal: Estimating the vector $\underline{s}$, up to some indeterminacies.

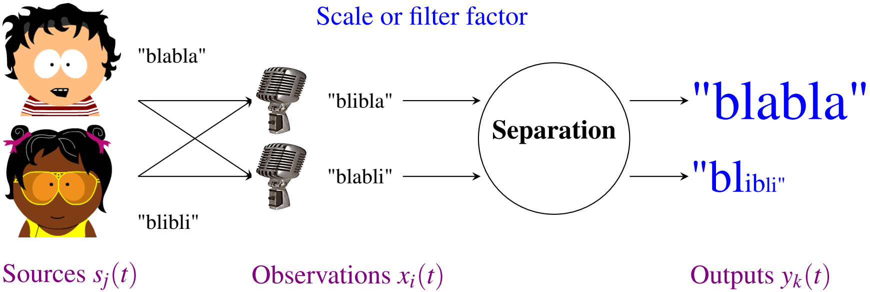


Blind Source Separation problem

- N unknown sources s_j.
- One unknown operator $\mathcal{A}$.
- P observed signals x_i with the global relation

$$\underline{x} = \mathcal{A}\left(\underline{s}\right).$$

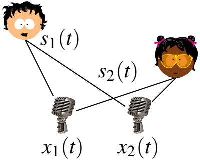
Goal: Estimating the vector $\underline{s}$, up to some indeterminacies.



Classes of mixtures

Most of the approaches process linear mixtures which are divided in three categories:

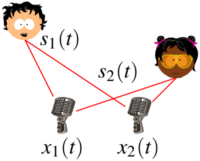
1. Linear instantaneous (LI) mixtures: $x_i(t) = \sum_{j=1}^{N} a_{ij}\, s_j(t)$



Classes of mixtures

Most of the approaches process linear mixtures which are divided in three categories:

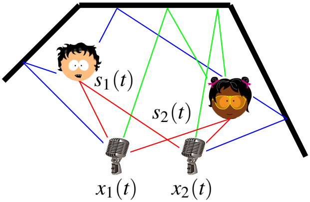
1. Linear instantaneous (LI) mixtures: $x_i(t) = \sum_{j=1}^{N} a_{ij}\, s_j(t)$
2. Attenuated and delayed (AD) mixtures: $x_i(t) = \sum_{j=1}^{N} a_{ij}\, s_j(t - t_{ij})$



Classes of mixtures

Most of the approaches process linear mixtures which are divided in three categories:

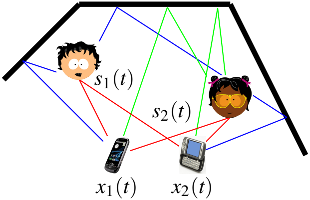
1. Linear instantaneous (LI) mixtures: $x_i(t) = \sum_{j=1}^{N} a_{ij} s_j(t)$
2. Attenuated and delayed (AD) mixtures: $x_i(t) = \sum_{j=1}^{N} a_{ij} s_j(t - t_{ij})$
3. Convolutive mixtures:
   $x_i(t) = \sum_{j=1}^{N} \sum_{k=-\infty}^{+\infty} a_{ijk} s_j(t - t_{ijk}) = \sum_{j=1}^{N} a_{ij}(t) * s_j(t)$



Classes of mixtures

Most of the approaches process linear mixtures which are divided in three categories:

1. Linear instantaneous (LI) mixtures: $x_i(t) = \sum_{j=1}^{N} a_{ij}\, s_j(t)$
2. Attenuated and delayed (AD) mixtures: $x_i(t) = \sum_{j=1}^{N} a_{ij}\, s_j(t - t_{ij})$
3. Convolutive mixtures:
   $x_i(t) = \sum_{j=1}^{N} \sum_{k=-\infty}^{+\infty} a_{ijk}\, s_j(t - t_{ijk}) = \sum_{j=1}^{N} a_{ij}(t) * s_j(t)$



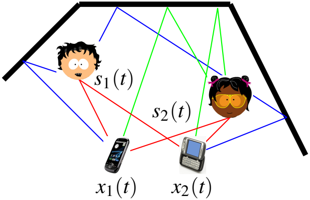
But more recently, interest for some problems with nonlinear mixtures:

4. Post-nonlinear (convolutive) mixtures: $x_i(t) = f_i\left(\sum_{j=1}^{N} a_{ij}(t) * s_j(t)\right)$

Classes of mixtures

Most of the approaches process linear mixtures which are divided in three categories:

1. Linear instantaneous (LI) mixtures: $x_i(t) = \sum_{j=1}^{N} a_{ij}\, s_j(t)$
2. Attenuated and delayed (AD) mixtures: $x_i(t) = \sum_{j=1}^{N} a_{ij}\, s_j(t - t_{ij})$
3. Convolutive mixtures:
   $x_i(t) = \sum_{j=1}^{N} \sum_{k=-\infty}^{+\infty} a_{ijk}\, s_j(t - t_{ijk}) = \sum_{j=1}^{N} a_{ij}(t) * s_j(t)$



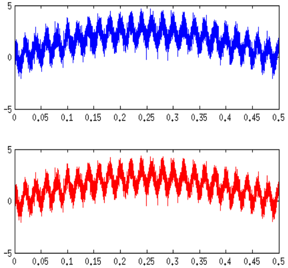
But more recently, interest for some problems with nonlinear mixtures:

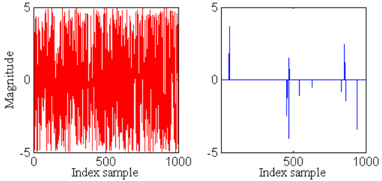
4. Post-nonlinear (convolutive) mixtures: $x_i(t) = f_i\left(\sum_{j=1}^{N} a_{ij}(t) * s_j(t)\right)$
5. Other classes of nonlinear mixtures (linear quadratic, polynomial, etc)

How to solve Blind Source Separation?

Three main families of methods:

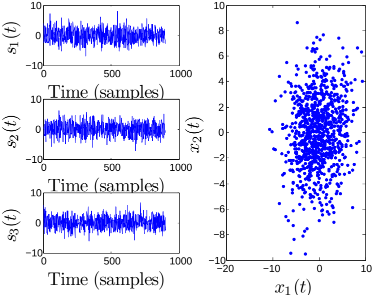
1. **Independent Component Analysis (ICA):** Sources are statistically independent, stationary and at most one of them is Gaussian (in their basic versions).
2. **Sparse Component Analysis (SCA):** Sparse sources (i.e. most of the samples are null (or close to zero)). Purpose of this seminar
3. **Non-negative Matrix Factorization (NMF):** Both sources et mixtures are positive, with possibly sparsity constraints.





Sparse Component Analysis (1)

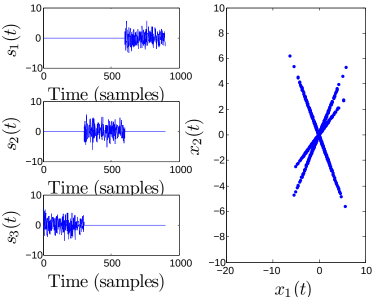
- Can process underdetermined mixtures (i.e. more sources than observations)



Sparse Component Analysis (1)

- Can process underdetermined mixtures (i.e. more sources than observations)
- If source signals are sparse in an analysis domain (e.g. time, Fourier, time-frequency, wavelet domains)

➪ $\forall t, \quad \exists k \in \{1 \ldots P\}, \quad \forall i \in \{1 \ldots N\}, \quad x_i(t) = \sum_{j=1}^{N} a_{ij}\, s_j(t) \simeq a_{ik}\, s_k(t)$



Sparsity assumptions:

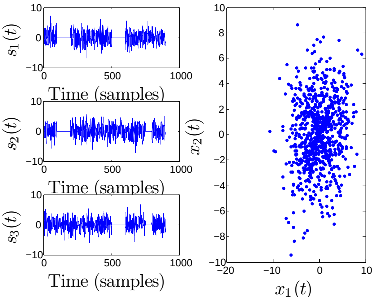
1. Sources are disjoint orthogonal (WDO)

➪ BSS $\simeq$ clustering problem (Jourjine *et al.*, 2000)

Sparse Component Analysis (1)

- Can process underdetermined mixtures (i.e. more sources than observations)
- If source signals are sparse in an analysis domain (e.g. time, Fourier, time-frequency, wavelet domains)

➪ $\forall t, \quad \exists k \in \{1 \ldots P\}, \quad \forall i \in \{1 \ldots N\}, \quad x_i(t) = \sum_{j=1}^{N} a_{ij}\, s_j(t) \simeq a_{ik}\, s_k(t)$



Sparsity assumptions:

1. Sources are disjoint orthogonal (WDO)

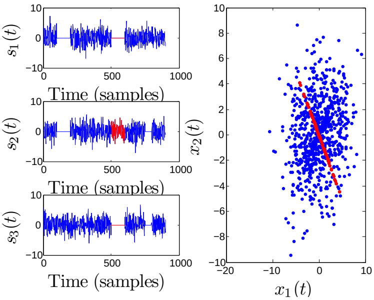
   ➪ BSS $\simeq$ clustering problem (Jourjine *et al.*, 2000)

2. Sources overlap, except if a few zones (to find) where only one of them is active (Deville *et al.*, 2001–2012)

Sparse Component Analysis (1)

- Can process underdetermined mixtures (i.e. more sources than observations)
- If source signals are sparse in an analysis domain (e.g. time, Fourier, time-frequency, wavelet domains)

➪ $\forall t, \quad \exists k \in \{1 \dots P\}, \quad \forall i \in \{1 \dots N\}, \quad x_i(t) = \sum_{j=1}^{N} a_{ij}\, s_j(t) \simeq a_{ik}\, s_k(t)$



Sparsity assumptions:

1. Sources are disjoint orthogonal (WDO)

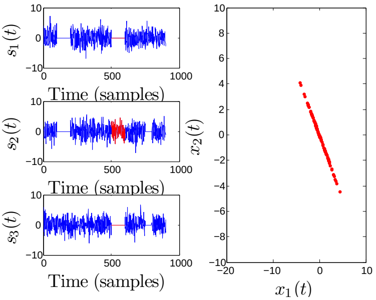
➪ BSS $\simeq$ clustering problem (Jourjine *et al.*, 2000)

2. Sources overlap, except if a few zones (to find) where only one of them is active (Deville *et al.*, 2001–2012)

Sparse Component Analysis (1)

- Can process underdetermined mixtures (i.e. more sources than observations)
- If source signals are sparse in an analysis domain (e.g. time, Fourier, time-frequency, wavelet domains)

➪ $\forall t, \quad \exists k \in \{1 \dots P\}, \quad \forall i \in \{1 \dots N\}, \quad x_i(t) = \sum_{j=1}^{N} a_{ij} s_j(t) \simeq a_{ik} s_k(t)$



Sparsity assumptions:

1. Sources are disjoint orthogonal (WDO)

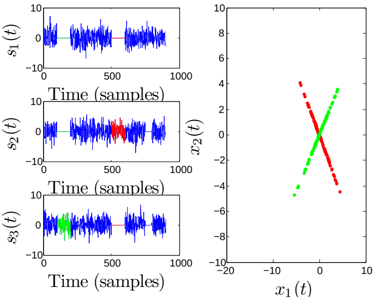
➪ BSS $\simeq$ clustering problem (Jourjine *et al.*, 2000)

2. Sources overlap, except if a few zones (to find) where only one of them is active (Deville *et al.*, 2001–2012)

Sparse Component Analysis (1)

- Can process underdetermined mixtures (i.e. more sources than observations)
- If source signals are sparse in an analysis domain (e.g. time, Fourier, time-frequency, wavelet domains)

➪ $\forall t, \quad \exists k \in \{1 \dots P\}, \quad \forall i \in \{1 \dots N\}, \quad x_i(t) = \sum_{j=1}^{N} a_{ij}\, s_j(t) \simeq a_{ik}\, s_k(t)$



Sparsity assumptions:

1. Sources are disjoint orthogonal (WDO)

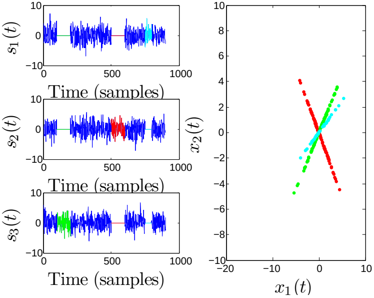
➪ BSS $\simeq$ clustering problem (Jourjine *et al.*, 2000)

2. Sources overlap, except if a few zones (to find) where only one of them is active (Deville *et al.*, 2001–2012)

Sparse Component Analysis (1)

- Can process underdetermined mixtures (i.e. more sources than observations)
- If source signals are sparse in an analysis domain (e.g. time, Fourier, time-frequency, wavelet domains)

➪ $\forall t, \quad \exists k \in \{1 \ldots P\}, \quad \forall i \in \{1 \ldots N\}, \quad x_i(t) = \sum_{j=1}^{N} a_{ij}\, s_j(t) \simeq a_{ik}\, s_k(t)$



Sparsity assumptions:

1. Sources are disjoint orthogonal (WDO)

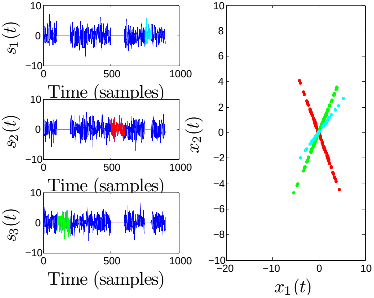
➪ BSS $\simeq$ clustering problem (Jourjine *et al.*, 2000)

2. Sources overlap, except if a few zones (to find) where only one of them is active (Deville *et al.*, 2001–2012)

Sparse Component Analysis (1)

- Can process underdetermined mixtures (i.e. more sources than observations)
- If source signals are sparse in an analysis domain (e.g. time, Fourier, time-frequency, wavelet domains)

➩ $\forall t, \quad \exists k \in \{1 \dots P\}, \quad \forall i \in \{1 \dots N\}, \quad x_i(t) = \sum_{j=1}^{N} a_{ij} \, s_j(t) \simeq a_{ik} \, s_k(t)$



Sparsity assumptions:

1. Sources are disjoint orthogonal (WDO)

   ➩ BSS $\simeq$ clustering problem (Jourjine *et al.*, 2000)
2. Sources overlap, except if a few zones (to find) where only one of them is active (Deville *et al.*, 2001–2012)
3. $2 \leq Q < P$ sources are always active at each time (Cichocki *et al.*, 2004–2006)

Sparse component analysis (2)

Structure

Most of the SCA approaches follow the same structure:

1. Jointly sparsifying the observations $x_i(t)$
2. Estimating the mixing parameters:
   - e.g. by finding single-source zones using the ratio of observations (Deville *et al.*, 2001–2006), correlation (Deville *et al.*, 2004–2012), PCA (Arberet *et al.*, 2006–2010), the real and imaginary parts of the observations (Reju *et al.*, 2010), etc...
   - clustering estimates of the mixing parameters in the above zones (DEMIX, Selective K-means and K-medians, etc)
3. Estimating the sources (as an inverse problem)

Applications

As BSS is generic, SCA methods may be applied in numerous applications:

Audio domain

- Source cancelation (karaoke-like application)

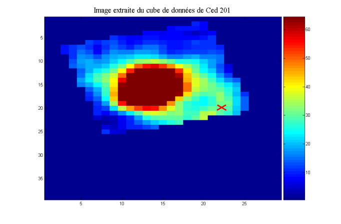
  Observation 1 Observation 2 Output "without singer"
- Separation for re-spatialization of the sound
  Example (SiSEC 2008): Observations Output 1 Output 2 Output 3
- Audio enhancement (improving the perceptual sound of a speaker by removing the surrounding noise)
  Example (SiSEC 2010): Observations Output 1 Output 2

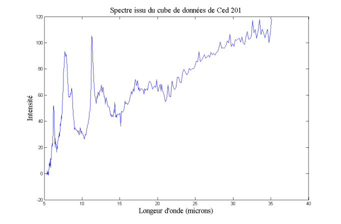
Applications

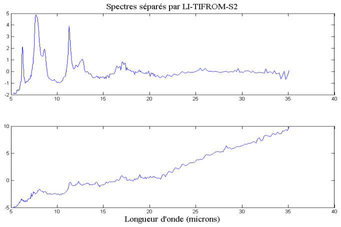
As BSS is generic, SCA methods may be applied in numerous applications:

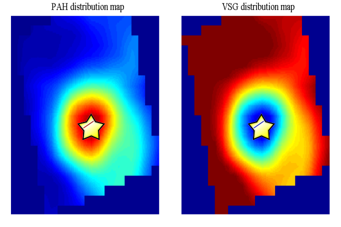
Astrophysical dust spectra from hyperspectral datacubes

From Berné *et al.* (2007–2009) and Puigt *et al.* (2009).





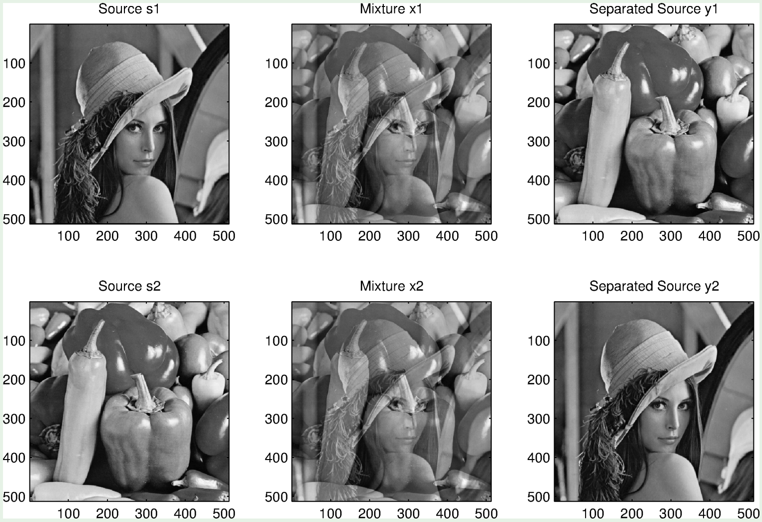




Applications

As BSS is generic, SCA methods may be applied in numerous applications:

Images (Meganem *et al.*, 2010)



Part I

Basic introduction to blind source separation

Part II

Post-nonlinear sparse component analysis

Part III

Real-time source localization: a brief introduction

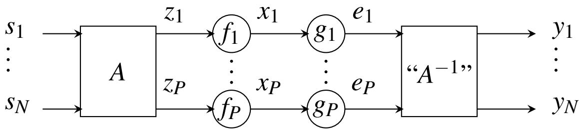
This work has been published in:

- M. Puigt, A. Griffin, and A. Mouchtaris, “Post-nonlinear speech mixture identification using single-source temporal zones & curve clustering”, in Proc. of EUSIPCO, pp. 1844-1848, 2011.
- M. Puigt, A. Griffin, and A. Mouchtaris, “Nonlinear blind mixture identification using local source sparsity and functional data clustering”, in Proc. of IEEE SAM, pp. 481-484, 2012.
- M. Puigt, A. Griffin, and A. Mouchtaris, “Post-nonlinear sparse component analysis using single-source zones and functional data clustering”, Preprint, `http://arxiv.org/abs/1204.1085`.

Post-nonlinear SCA: motivation

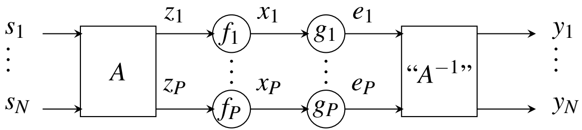
- Most of the people use their mobile device (e.g. smartphone) as a notebook: capturing images, videos, sounds and sharing them (social networks, streaming video websites, etc)
- But mobile devices are *mobile*:
  - Cheap and small microphone(s) & loudspeakers
  - ➩ provide nonlinearities in sound recording & restitution
  - Example: live recordings (ex: found on Youtube)
  - Observed sound signals may be written as post-nonlinear (PNL) convolutive mixtures of source signals: $x_i(t) = f_i\left(\sum_{j=1}^{N} a_{ij}(t) * s_j(t)\right)$
- To takkle this problem, the litterature provides:
  1. several ICA methods (limited to determined mixtures)
  2. a few SCA methods requiring the WDO assumption (Theis & Amari, 2004, van Vaerenbergh & Santamaría, 2006)
  - ➩ As we know that SCA outperforms ICA (Deville & Puigt, 2007), how to extend SCA methods to PNL convolutive mixtures?
  - ➩ In a first stage, how to extend LI-SCA methods to (instantaneous) PNL mixtures $x_i(t) = f_i\left(\sum_{j=1}^{N} a_{ij} \cdot s_j(t)\right)$?

Structure of the proposed method(s)



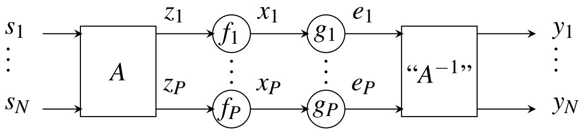
- Mirror structure in PNL-ICA ($P \geq N$)
  1. Estimate the inverse g_i of the NL functions f_i
  2. Linearize the mixtures
  3. Estimate A^{-1} and deduce the sources

Structure of the proposed method(s)



- Mirror structure in PNL-ICA ($P \geq N$)
  1. Estimate the inverse g_i of the NL functions f_i
  2. Linearize the mixtures
  3. Estimate A^{-1} and deduce the sources
- Mirror structure in PNL-SCA ($P \geq N$ or $P < N$)
  1. Estimate the NL functions f_i
     1. Cut $x_i(t)$ in temporal analysis zones $\mathcal{T}$
     2. Find temporal single-source zones
     3. Estimate NL mappings
  2. Linearize the mixtures (as an inverse problem)
  3. Estimate A (e.g. apply a LI-SCA method)
  4. Estimate the sources (as an inverse problem)

Structure of the proposed method(s)



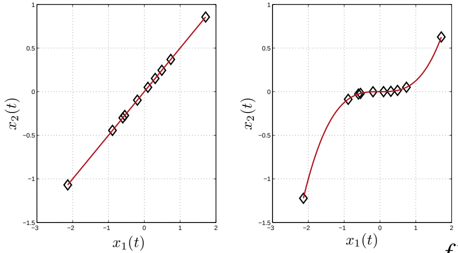
- Mirror structure in PNL-ICA ($P \geq N$)
  1. Estimate the inverse g_i of the NL functions f_i
  2. Linearize the mixtures
  3. Estimate A^{-1} and deduce the sources
- Mirror structure in PNL-SCA ($P \geq N$ or $P < N$)
  1. Estimate the NL functions f_i
     1. Cut $x_i(t)$ in temporal analysis zones $\mathcal{T}$
     2. Find temporal single-source zones
     3. Estimate NL mappings
  2. Linearize the mixtures (as an inverse problem)
  3. Estimate A (e.g. apply a LI-SCA method)
  4. Estimate the sources (as an inverse problem)

What about the sparsifying transform?

Not a good idea (not sparser observations) ➩ application: speech signals

Geometrical point of view

Let us imagine that, in one zone, only one source, say s_k, is active...



$$\forall i \in \{1, \ldots, P\} \qquad x_i(t) = f_i\left(a_{ik} s_k(t)\right) \Rightarrow s_k(t) = \frac{f_i^{-1}\left(x_i(t)\right)}{a_{ik}}$$

and

$$x_i(t) = f_i\left(\frac{a_{ik}}{a_{1k}} f_1^{-1}\left(x_1(t)\right)\right) = \phi_{ik}(x_1(t))$$

Questions:

- How to find single-source zones?
- How to estimate ϕ_{ik}?

Single-source confidence measures (1)

Mutual information (Puigt *et al.*, 2011)

- In linear mixtures, a source is isolated iff observations are proportional (correlation – Deville & Puigt, 2007)
- In PNL mixtures, need to measure the nonlinear correlation between observations.
- Done using mutual information (Dionisio *et al.*, 2004)

$$I(\underline{x}) = -\mathbb{E}\left\{\log \frac{\prod_{i=1}^{P} \mathbb{P}_{x_i}(x_i)}{\mathbb{P}_{\underline{x}}(\underline{x})}\right\}$$

which may be normalised as $I_{\text{norm}}(\underline{x}) = \sqrt{1 - e^{-2\,I(\underline{x})}}$.

➯ This implies that sources are mutually independent (✔ for speech—Puigt *et al.*, 2009)

Single-source confidence measures (2)

Manifold learning based measures (Puigt *et al.*, 2012a)

- Alternative to mutual information
- If the NL functions are smooth, then ϕ_{ik} are smooth and locally linear

➪ We can *locally* apply linear measures (Manifold learning)

- Linear Tangent Space Approximation (LTSA—van der Maaten *et al.*, 2009) approximates the manifold around a value by its tangent in this point

1. We consider each point t_i of a zone $\mathcal{T}$ and we find its K-NN
2. We apply a linear SSCM in this neighbourhood
   - correlation (Deville & Puigt, 2007): $C_{x_1,x_j}(t_i)$
   - ratio of eigenvalues (Arberet *et al.*, 2010): $R(t_i) = \frac{\lambda_1(t_i)}{\sum_{j=1}^{K} \lambda_j(t_i)}$

➪ *Global* single-source confidence measure as the geometric mean of all the local SSCMs, respectively denoted $\mathcal{C}(\underline{x})$ and $\mathcal{R}(\underline{x})$

Single-source confidence measures (3)

Finally...

We look for zones $\mathcal{T}$ such that $\text{SSCM}(\underline{x}) > 1 - \varepsilon_1$

Case of non-null unactive sources

- Problem may appear if inactive sources are constant but non-zero as $x_i(t) = f_i\left(a_{ik}s_k(t) + \alpha_i(T)\right)$ where $\alpha_i(T) = \sum_{j \neq k} a_{ij}\overline{s_j}$
- Not a problem in Linear mixtures where observations may be centered...

➥ If all $f_i(0) = 0$ (not limiting assumption ✔), we discard all the estimated curves $\widehat{\phi}_{ik}$ which do not satisfy $\left|\widehat{\phi}_{ik}(0)\right| < \varepsilon_2$.

Functional data analysis and clustering (1)

- Different single-source zones may lead to scattered functions associated with the *same* source
- Interest to cluster all these zones in order to get an accurate estimate of the nonlinear mappings ϕ_{ik}
- Previously proposed methods also clustered data:
  1. Theis and Amari: geometrical preprocessing sensitive to noise or non-ideal single source zones
  2. Van Vaerenbergh and Santamaría: spectral clustering with curve distances limitations and which does not allow the curves to intersect (while all $f_i(0) = 0$) $\Rightarrow$ additional curve shape assumptions
- We propose of taking advantage of the single-source zones: we estimate some parameters which adequately describe each scattered function and we cluster them.
- In particular, we test two families of methods:
  1. We use a classical functional data clustering method
  2. We propose a new approach well-suited to the considered problem.

Functional data analysis and clustering (2)

Filtering functional data clustering

Based on B-spline approximation (Abraham *et al.*, 2003): we approximate any nonlinear function with respect to a basis of polynomials

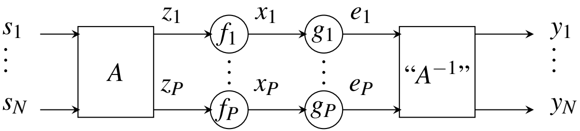
1. Select all the single-source zones and fix some knots locations
2. In each above zone, estimate $\widehat{\phi}_{ik}$ using B-splines (all the B-spline coefficients have the same meaning and describe each curve shape)
3. Cluster the B-spline coefficients, in order to cluster the curves (K-medians)

Proposed method (Puigt *et al.*, 2012a)

Based on manifold learning around 0

1. For each zone, find the K-NN points around 0
2. Estimate the linear DOA in each zone with an approach of the litterature (e.g. based on correlation or PCA)
3. Cluster these estimated DOAs (K-medians)

Final curve estimation and next steps of the separation approach



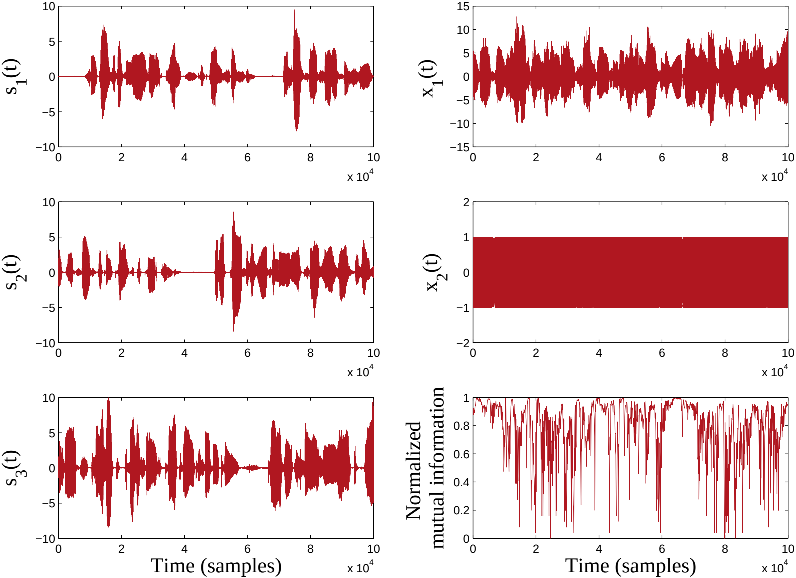
- We finally got separated functions that we e.g. can estimate thanks to B-splines, with much more knots and not-fixed locations.
- We can apply one approach of the literature to invert the nonlinearities (change the curves in lines).
- We then get a linear problem for which the mixing parameters can be estimated (slopes of the obtained lines).
- But we only focused on the nonlinearities estimation because *“it is of major importance for solving the BSS problem.”*
- Let us see an example!

Example

- $N = 3$ sources (5 s, $F_s = 20$ kHz, silent parts) and $P = 2$ sensors
- PNL mixture: $A = \begin{bmatrix} 1 & 1 & 0.9 \\ -0.9 & 0.5 & 1 \end{bmatrix}$ and $\begin{cases} f_1(t) & = & \tanh(t) + t \\ f_2(t) & = & \tanh(10t) \end{cases}$
- Mutual information estimation with 100 samples per analysis zone, $\varepsilon_1 = 0.01$, $\varepsilon_2 = 0.1$

Example

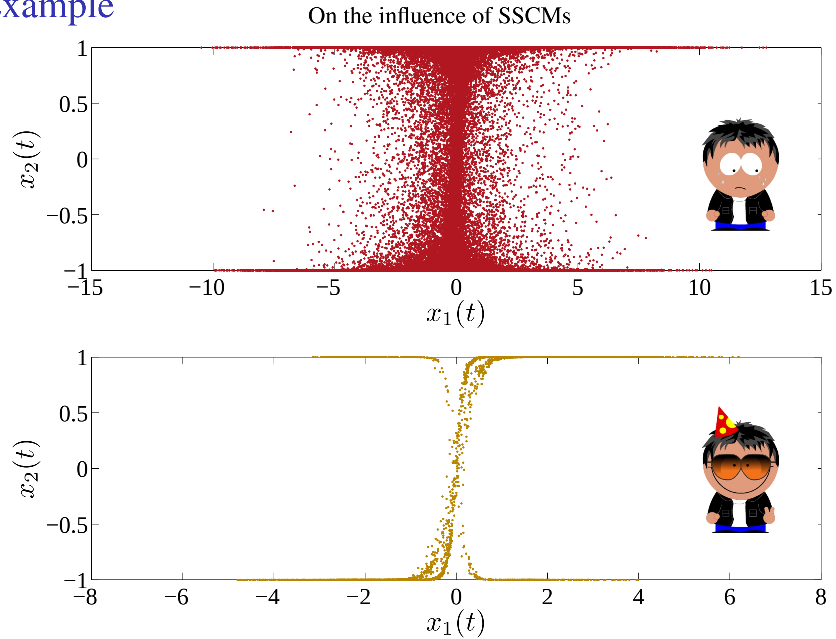
Despite the strong NL, we find single-source zones



Example

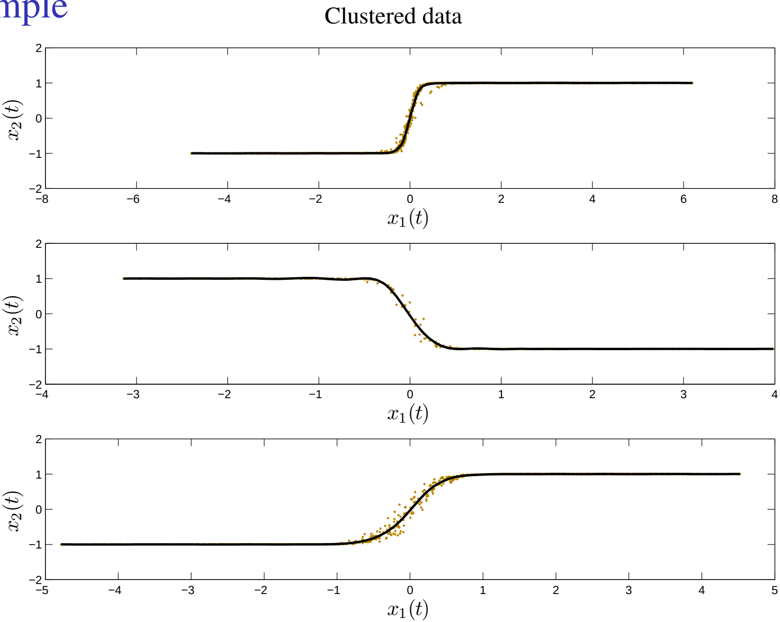
- $N = 3$ sources (5 s, $F_s = 20$ kHz, silent parts) and $P = 2$ sensors
- PNL mixture: $A = \begin{bmatrix} 1 & 1 & 0.9 \\ -0.9 & 0.5 & 1 \end{bmatrix}$ and $\begin{cases} f_1(t) & = & \tanh(t) + t \\ f_2(t) & = & \tanh(10t) \end{cases}$
- Mutual information estimation with 100 samples per analysis zone, $\varepsilon_1 = 0.01$, $\varepsilon_2 = 0.1$
- FDC technique: $\xi_i = -1.5 + 0.3i$ for $i \in \{0, \ldots, 10\}$ with B-splines of degree 4

Example



On the influence of SSCMs
1
0.5
0
−0.5
−1
−15
−10
−5
0
5
10
15
$x_1(t)$
$x_2(t)$
1
0.5
0
−0.5
−1
−8
−6
−4
−2
0
2
4
6
8
$x_1(t)$
$x_2(t)$

Example



Clustered data
2
1
0
−1
−2
−8
−6
−4
−2
0
2
4
6
8
$x_2(t)$
$x_1(t)$
2
1
0
−1
−2
−4
−3
−2
−1
0
1
2
3
4
$x_2(t)$
$x_1(t)$
2
1
0
−1
−2
−5
−4
−3
−2
−1
0
1
2
3
4
5
$x_2(t)$
$x_1(t)$

Example

- $N = 3$ sources (5 s, $F_s = 20$ kHz, silent parts) and $P = 2$ sensors
- PNL mixture: $A = \begin{bmatrix} 1 & 1 & 0.9 \\ -0.9 & 0.5 & 1 \end{bmatrix}$ and $\begin{cases} f_1(t) & = & \tanh(t) + t \\ f_2(t) & = & \tanh(10t) \end{cases}$
- Mutual information estimation with 100 samples per analysis zone, $\varepsilon_1 = 0.01$, $\varepsilon_2 = 0.1$
- FDC technique: $\xi_i = -1.5 + 0.3i$ for $i \in \{0, \ldots, 10\}$ with B-splines of degree 4
- Accurate classification and estimation, MSE: 2.5e-4, 5.3e-5, and 2.1e-5

Example

- $N = 3$ sources (5 s, $F_s = 20$ kHz, silent parts) and $P = 2$ sensors
- PNL mixture: $A = \begin{bmatrix} 1 & 1 & 0.9 \\ -0.9 & 0.5 & 1 \end{bmatrix}$ and $\begin{cases} f_1(t) & = & \tanh(t) + t \\ f_2(t) & = & \tanh(10t) \end{cases}$
- Mutual information estimation with 100 samples per analysis zone, $\varepsilon_1 = 0.01$, $\varepsilon_2 = 0.1$
- FDC technique: $\xi_i = -1.5 + 0.3i$ for $i \in \{0, \ldots, 10\}$ with B-splines of degree 4
- Accurate classification and estimation, MSE: 2.5e-4, 5.3e-5, and 2.1e-5

Detailed performance of the proposed approaches (Puigt *et al.*, 2012b)

We performed many tests (several NL functions, several mixing matrices while method's parameters vary) which show that:

- $I_{\text{norm}}(\underline{x})$ and $\mathcal{C}(\underline{x})$ are better-suited than $\mathcal{R}(\underline{x})$
- Our manifold-learning clustering approaches are more flexible than B-spline functional data clustering one

Conclusion and future work

- A general framework for extending linear SCA to PNL mixtures (and even more general NL mixtures, see Puigt *et al.*, 2012a)
- Estimation of the non-linearities combines single-source zones with functional data clustering
- We proposed some Manifold-learning-based techniques for both tasks
- We also tested classical measures (mutual information and B-splines functional data clustering)
- Our results show that our approaches allow an accurate estimation of the nonlinearities
- We still have to invert them
  - May be done with an approach of the litterature
  - Or with a future proposed method...
- Our approach restricted to signals which are sparse in the time domain
  ➪ not well-suited to music
- Still need to investigate PNL convolutive mixtures

Part I

Basic introduction to blind source separation

Part II

Post-nonlinear sparse component analysis

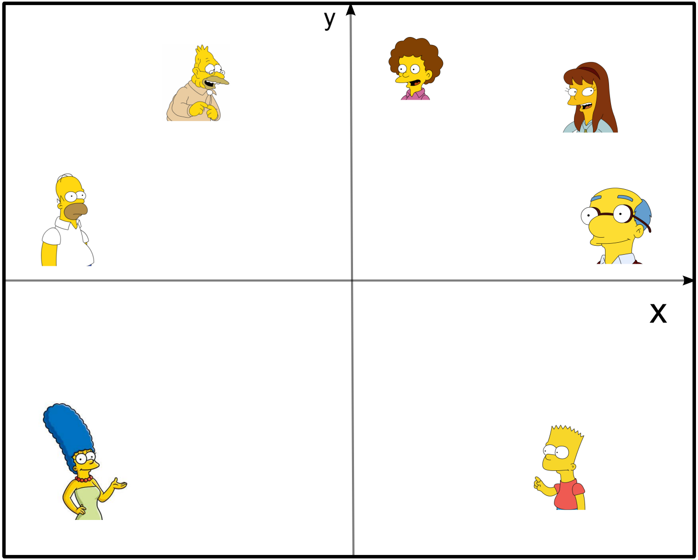
Part III

Real-time source localization: a brief introduction

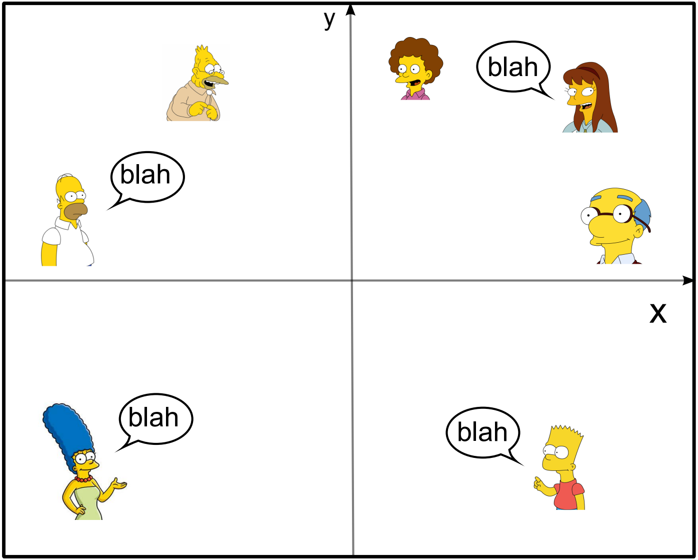
This work has been published in:

- D. Pavlidi, M. Puigt, A. Griffin, A. Mouchtaris, “Real-time multiple sound source localization using a circular microphone array based on single-source confidence measures”, in Proc. of IEEE ICASSP, pp. 2625-2628, 2012.
- D. Pavlidi, A. Griffin, M. Puigt, A. Mouchtaris, “Source counting in real-time sound source localization using a circular microphone array”, in Proc. of IEEE SAM, pp. 521-524, 2012.
- A. Griffin, D. Pavlidi, M. Puigt, A. Mouchtaris, “Real-time multiple speaker DOA estimation in a circular microphone array based on matching pursuit”, in Proc. of EUSIPCO, 2012.
- Journal paper just submitted to IEEE TASLP

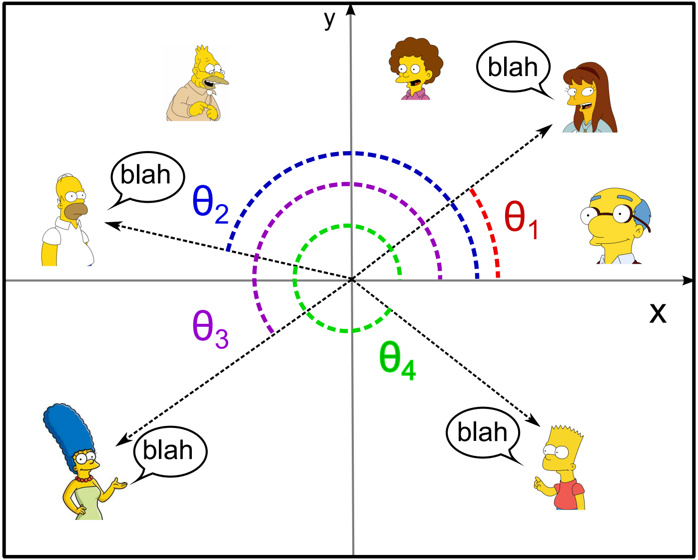
Motivation



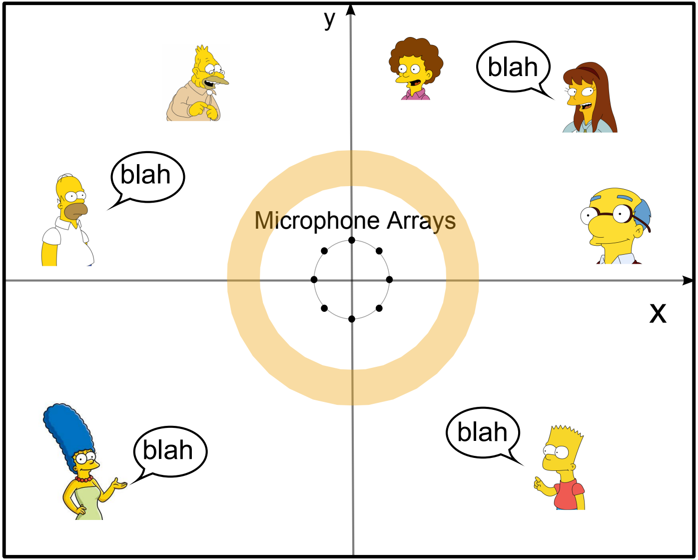
Motivation



Motivation



Motivation

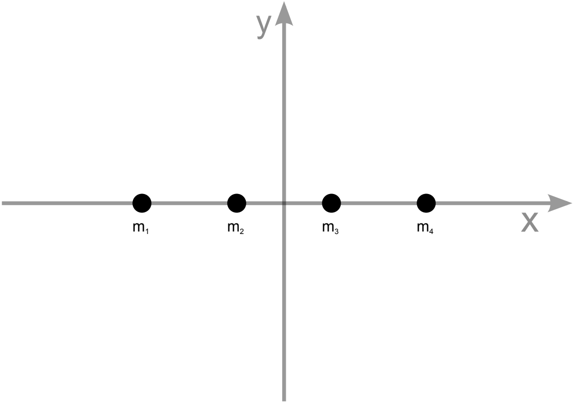


Microphone arrays

Geometry

Linear arrays

- Simple...

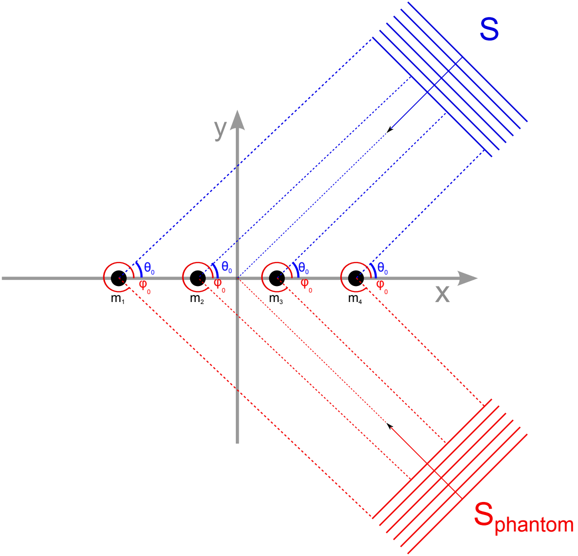


Microphone arrays

Geometry

Linear arrays

- Simple...
- But ambiguous!

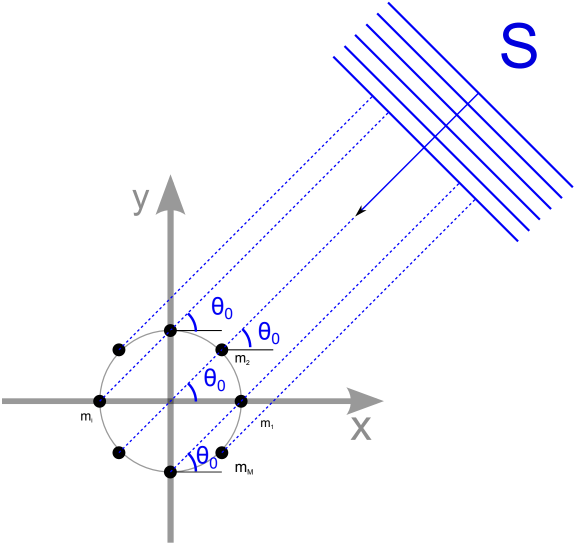


Microphone arrays

Geometry

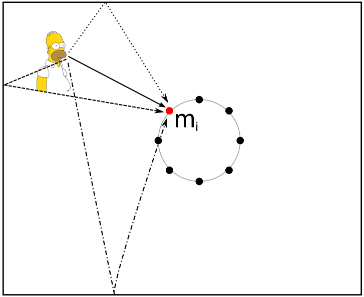
Circular arrays

Remove ambiguities!



Microphone arrays

Propagation Models



Reverberant Model

Time domain

$$x_i(t) = \sum_{g=1}^{P} h_{ig}(t) * s_g(t) + n_i(t)$$

Time-Frequency (TF) domain

$$X_i(t,\omega) = \sum_{g=1}^{P} H_{ig}(\omega) \cdot S_g(t,\omega) + N_i(t,\omega)$$

State of the Art

Single source localization

Based on Time difference of arrival

- the GCC family [Knapp 1976]

State of the Art

Single source localization

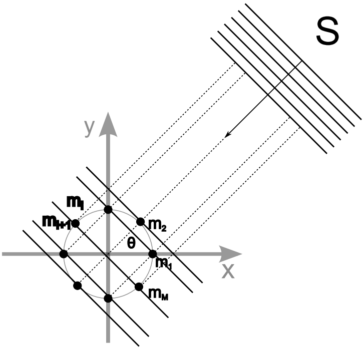
Based on Time difference of arrival

- the GCC family [Knapp 1976]

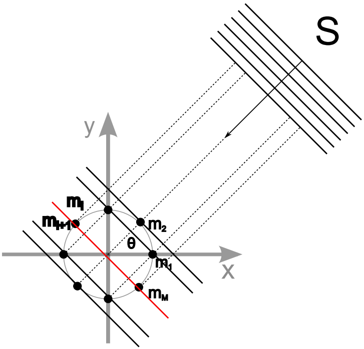
Multiple source localization

1. Based on statistics
   - beamforming (2nd order statistics — e.g. multiple signal classification (MUSIC))[Argentieri 2007]
   - Independent component analysis (2nd order or higher-order statistics) [Lombard 2008]
2. Using the sparsity paradigm
   - Sparse component analysis (SCA) with W-disjoint orthogonality [Swartling 2006]
   - Sparse component analysis with single-source confidence measure (framework of the present work)

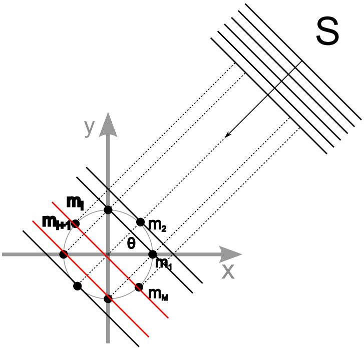
Single sound source localization using a circular microphone array [Karbasi 2007]



Single sound source localization using a circular microphone array [Karbasi 2007]

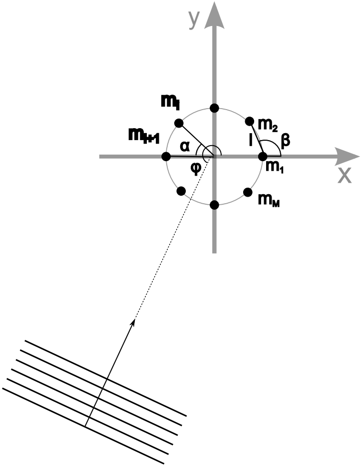


Single sound source localization using a circular microphone array [Karbasi 2007]



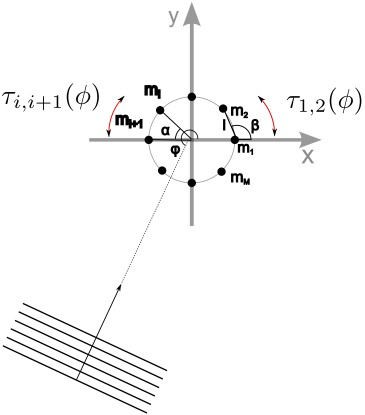
- $\tau_{i,i+1}(\theta)$
- $G^{\theta}_{i,i+1}(\omega) = \angle R_{i,i+1}(\omega) = e^{-j\omega\tau_{i,i+1}(\theta)}$

Single sound source localization using a circular microphone array [Karbasi 2007]



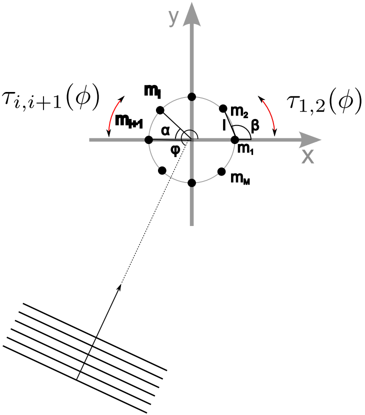
- $\tau_{i,i+1}(\phi) = \frac{l \sin(\beta - \phi + (i-1)\alpha)}{c}$
- $G^{\theta}_{i,i+1}(\omega) = \angle R_{i,i+1}(\omega) = e^{-j\omega\tau_{i,i+1}(\theta)}$

Single sound source localization using a circular microphone array [Karbasi 2007]



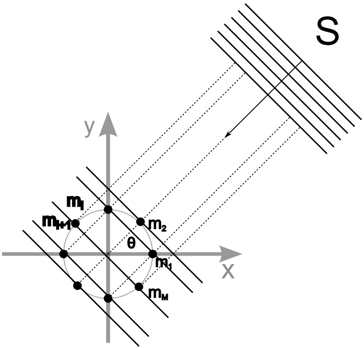
- $\tau_{i,i+1}(\phi) = \frac{l \sin(\beta - \phi + (i-1)\alpha)}{c}$
- $G^{\theta}_{i,i+1}(\omega) = \angle R_{i,i+1}(\omega) = e^{-j\omega\tau_{i,i+1}(\theta)}$
- Phase rotation factor: $\mathrm{PRF}_{m_i \to m_1}(\phi) = e^{-j\omega(\tau_{1,2}(\phi) - \tau_{i,i+1}(\phi))}$

Single sound source localization using a circular microphone array [Karbasi 2007]



- $\tau_{i,i+1}(\phi) = \frac{l\sin(\beta-\phi+(i-1)\alpha)}{c}$
- $G^{\theta}_{i,i+1}(\omega) = \angle R_{i,i+1}(\omega) = e^{-j\omega\tau_{i,i+1}(\theta)}$
- Phase rotation factor: $\text{PRF}_{m_i\to m_1}(\phi) = e^{-j\omega(\tau_{1,2}(\phi)-\tau_{i,i+1}(\phi))}$
- Circular integrated cross spectrum: $\text{CICS}(\phi) \triangleq \sum_{i=1}^{M} \text{PRF}_{m_i\to m_1}(\phi) G^{\theta}_{i,i+1}(\omega)$

Single sound source localization using a circular microphone array [Karbasi 2007]

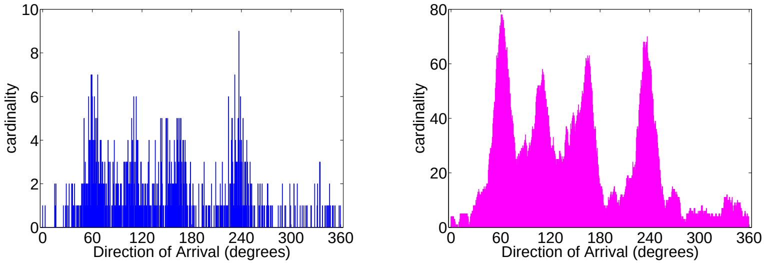


- $\tau_{i,i+1}(\phi) = \frac{l \sin(\beta - \phi + (i-1)\alpha)}{c}$
- $G^{\theta}_{i,i+1}(\omega) = \angle R_{i,i+1}(\omega) = e^{-j\omega\tau_{i,i+1}(\theta)}$
- Phase rotation factor:
  $\mathrm{PRF}_{m_i \to m_1}(\phi) = e^{-j\omega(\tau_{1,2}(\phi) - \tau_{i,i+1}(\phi))}$
- Circular integrated cross spectrum:
  $$\mathrm{CICS}(\phi) \triangleq \sum_{i=1}^{M} \mathrm{PRF}_{m_i \to m_1}(\phi) G^{\theta}_{i,i+1}(\omega)$$

$$\hat{\theta} = \arg \max_{0 \leq \phi < 2\pi} |\mathrm{CICS}(\phi)|.$$

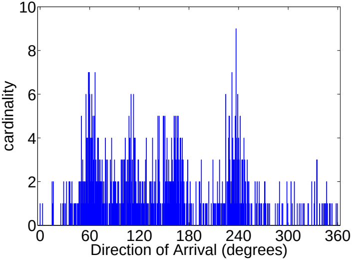
Multiple Sound Source localization method

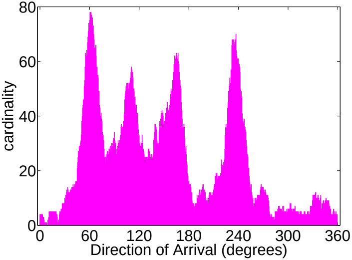
- If we combine the linear SSCM (e.g. the correlation) with the above single-source localization approach, we provide a real-time multiple-source localization method (Pavlidi *et al.*, 2012a).
- Algorithm:
  1. We consider an history length of the signal (typically 1 s)
  2. We cut it in frames (typically 2048 points)
  3. We compute a FFT on each frame
  4. We find Constant-time single-source zones (Puigt & Deville, 2007)
  5. We estimate the associated DOA in each of these zones
  6. We derive a (smoothed) histogram of the above DOAs



Multiple Sound Source localization method

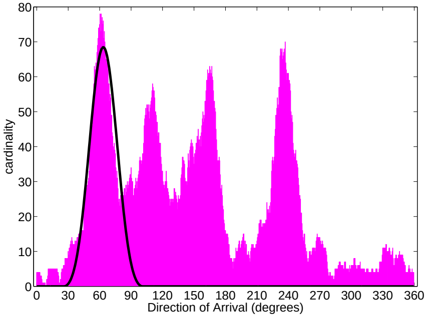
- If we combine the linear SSCM (e.g. the correlation) with the above single-source localization approach, we provide a real-time multiple-source localization method (Pavlidi *et al.*, 2012a).
- Algorithm:
  1. We consider an history length of the signal (typically 1 s)
  2. We cut it in frames (typically 2048 points)
  3. We compute a FFT on each frame
  4. We find Constant-time single-source zones (Puigt & Deville, 2007)
  5. We estimate the associated DOA in each of these zones
  6. We derive a (smoothed) histogram of the above DOAs
  7. We count the number of sources and we estimate their actual DOAs





Source counting

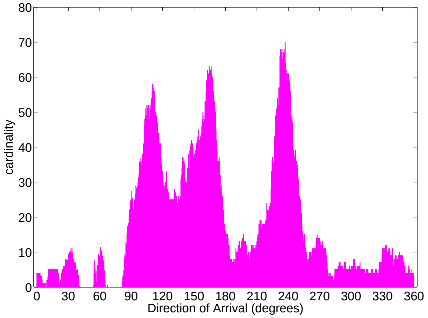
- We proposed 3 methods for source counting (Pavlidi *et al.*, 2012b)
- But only the most efficient one is here described
- Based on Matching Pursuit (model the histogram as linear combination of pulses)



- Define $\gamma = [\gamma_i], i = 1, \cdots P_{\mathrm{MAX}}$.
- Correlate the source atom with the histogram
- Detect highest peak, set $i = 1$
- Calculate its contribution: $\delta_i = \sum_j \frac{y_{i,j} - y_{i+1,j}}{y_{1,j}}$
- If $\delta_i \geq \gamma_i$, remove it, increment i.
- Continue iteratively until $\delta_i < \gamma_i$ (or $i = P_{\mathrm{MAX}}$)

Source counting

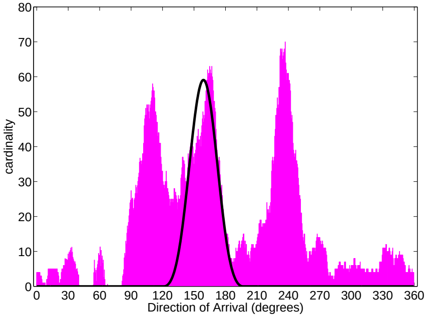
- We proposed 3 methods for source counting (Pavlidi *et al.*, 2012b)
- But only the most efficient one is here described
- Based on Matching Pursuit (model the histogram as linear combination of pulses)



- Define $\gamma = [\gamma_i], i = 1, \cdots P_{\text{MAX}}$.
- Correlate the source atom with the histogram
- Detect highest peak, set $i = 1$
- Calculate its contribution: $\delta_i = \sum_j \frac{y_{i,j} - y_{i+1,j}}{y_{1,j}}$
- If $\delta_i \geq \gamma_i$, remove it, increment i.
- Continue iteratively until $\delta_i < \gamma_i$ (or $i = P_{\text{MAX}}$)

Source counting

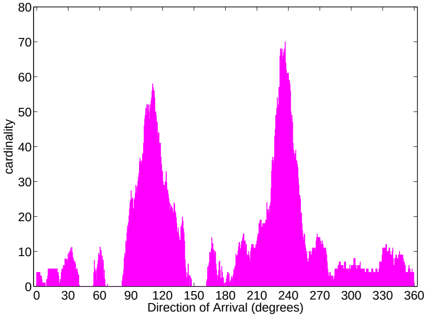
- We proposed 3 methods for source counting (Pavlidi *et al.*, 2012b)
- But only the most efficient one is here described
- Based on Matching Pursuit (model the histogram as linear combination of pulses)



- Define $\gamma = [\gamma_i], i = 1, \cdots P_{\text{MAX}}$.
- Correlate the source atom with the histogram
- Detect highest peak, set $i = 1$
- Calculate its contribution: $\delta_i = \sum_j \frac{y_{i,j} - y_{i+1,j}}{y_{1,j}}$
- If $\delta_i \geq \gamma_i$, remove it, increment i.
- Continue iteratively until $\delta_i < \gamma_i$ (or $i = P_{\text{MAX}}$)

Source counting

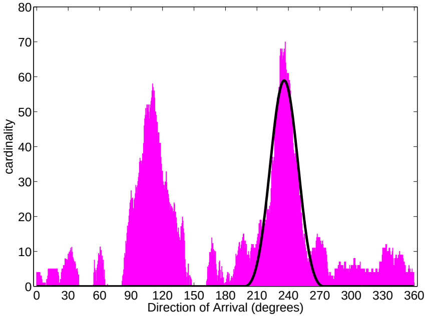
- We proposed 3 methods for source counting (Pavlidi *et al.*, 2012b)
- But only the most efficient one is here described
- Based on Matching Pursuit (model the histogram as linear combination of pulses)



- Define $\gamma = [\gamma_i], i = 1, \cdots P_{\text{MAX}}$.
- Correlate the source atom with the histogram
- Detect highest peak, set $i = 1$
- Calculate its contribution: $\delta_i = \sum_j \frac{y_{i,j} - y_{i+1,j}}{y_{1,j}}$
- If $\delta_i \geq \gamma_i$, remove it, increment i.
- Continue iteratively until $\delta_i < \gamma_i$ (or $i = P_{\text{MAX}}$)

Source counting

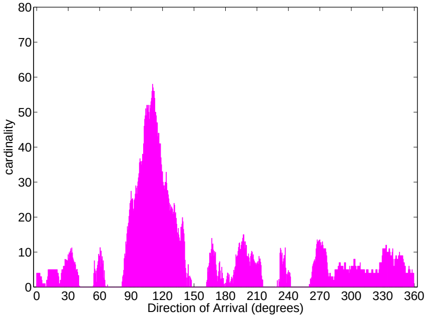
- We proposed 3 methods for source counting (Pavlidi *et al.*, 2012b)
- But only the most efficient one is here described
- Based on Matching Pursuit (model the histogram as linear combination of pulses)



- Define $\gamma = [\gamma_i], i = 1, \cdots P_{\text{MAX}}$.
- Correlate the source atom with the histogram
- Detect highest peak, set $i = 1$
- Calculate its contribution: $\delta_i = \sum_j \frac{y_{i,j} - y_{i+1,j}}{y_{1,j}}$
- If $\delta_i \geq \gamma_i$, remove it, increment i.
- Continue iteratively until $\delta_i < \gamma_i$ (or $i = P_{\text{MAX}}$)

Source counting

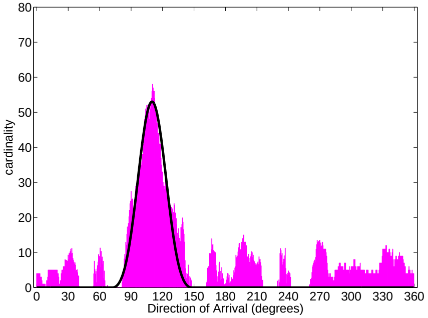
- We proposed 3 methods for source counting (Pavlidi *et al.*, 2012b)
- But only the most efficient one is here described
- Based on Matching Pursuit (model the histogram as linear combination of pulses)



- Define $\gamma = [\gamma_i], i = 1, \cdots P_{\text{MAX}}$.
- Correlate the source atom with the histogram
- Detect highest peak, set $i = 1$
- Calculate its contribution: $\delta_i = \sum_j \frac{y_{i,j} - y_{i+1,j}}{y_{1,j}}$
- If $\delta_i \geq \gamma_i$, remove it, increment i.
- Continue iteratively until $\delta_i < \gamma_i$ (or $i = P_{\text{MAX}}$)

Source counting

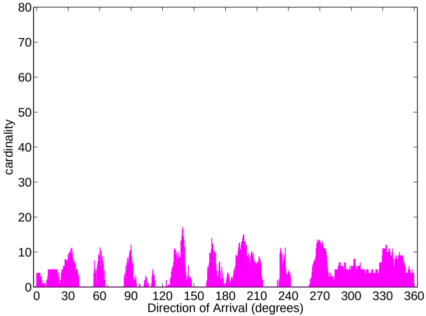
- We proposed 3 methods for source counting (Pavlidi *et al.*, 2012b)
- But only the most efficient one is here described
- Based on Matching Pursuit (model the histogram as linear combination of pulses)



- Define $\gamma = [\gamma_i], i = 1, \cdots P_{\text{MAX}}$.
- Correlate the source atom with the histogram
- Detect highest peak, set $i = 1$
- Calculate its contribution: $\delta_i = \sum_j \frac{y_{i,j} - y_{i+1,j}}{y_{1,j}}$
- If $\delta_i \geq \gamma_i$, remove it, increment i.
- Continue iteratively until $\delta_i < \gamma_i$ (or $i = P_{\text{MAX}}$)

Source counting

- We proposed 3 methods for source counting (Pavlidi *et al.*, 2012b)
- But only the most efficient one is here described
- Based on Matching Pursuit (model the histogram as linear combination of pulses)

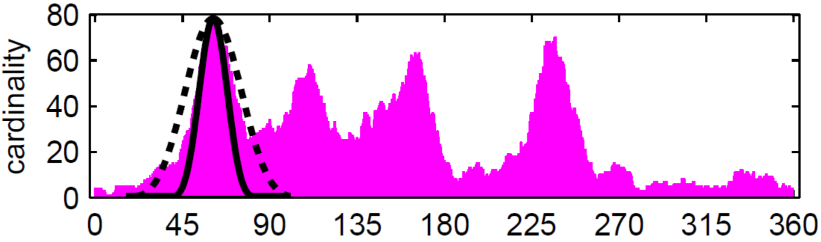


- Define $\gamma = [\gamma_i], i = 1, \cdots P_{\mathrm{MAX}}$.
- Correlate the source atom with the histogram
- Detect highest peak, set $i = 1$
- Calculate its contribution: $\delta_i = \sum_j \frac{y_{i,j} - y_{i+1,j}}{y_{1,j}}$
- If $\delta_i \geq \gamma_i$, remove it, increment i.
- Continue iteratively until $\delta_i < \gamma_i$ (or $i = P_{\mathrm{MAX}}$)

Final DOA estimation

- Two proposed methods: taking the indices of the highest value of the peaks or using Matching Pursuit (Griffin *et al.*, 2012)
- Key idea: The width of the above pulses is really important for getting accurate estimates (its shape may vary with the mixing conditions).
- Ideally, its optimal width should be estimated from the histogram (but time consumming)

➪ Combine two widths (was shown to be an acceptable trade-off)



➪ The whole method runs 55% real-time!

Performance (1)

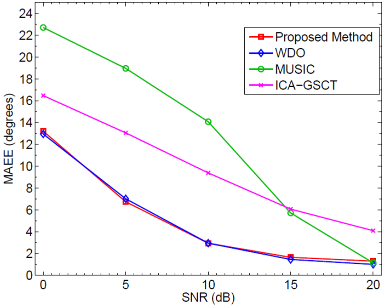
DOA estimation accuracy

- Simulations (6 sources, additive white noise, $T_{60} = 0.25$ s, room size: $4\text{m} \times 6\text{m} \times 3\text{m}$, 8 microphones) and comparison with Wideband Music, ICA–GSCT, and WDO-based method.
- Our approach is the less computational demanding (2.6E6 vs. between 3.9E6 & 35E6 operations)...

Performance (1)

DOA estimation accuracy

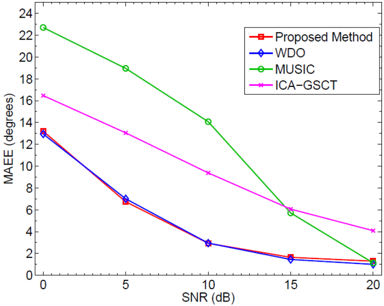
- Simulations (6 sources, additive white noise, $T_{60} = 0.25$ s, room size: 4m $\times$ 6m $\times$ 3m, 8 microphones) and comparison with Wideband Music, ICA–GSCT, and WDO-based method.
- Our approach is the less computational demanding (2.6E6 vs. between 3.9E6 & 35E6 operations)...
- But it outperforms all the methods of the litterature except the WDO one (perf. criterion: MAEE)

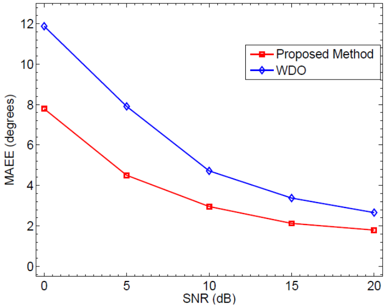


Performance (1)

DOA estimation accuracy

- Simulations (6 sources, additive white noise, $T_{60} = 0.25$ s, room size: 4m $\times$ 6m $\times$ 3m, 8 microphones) and comparison with Wideband Music, ICA–GSCT, and WDO-based method.
- Our approach is the less computational demanding (2.6E6 vs. between 3.9E6 & 35E6 operations)...
- But it outperforms all the methods of the litterature except the WDO one (perf. criterion: MAEE)
- It outperforms WDO in a 2-sources scenario





Performance (2)

Source counting accuracy

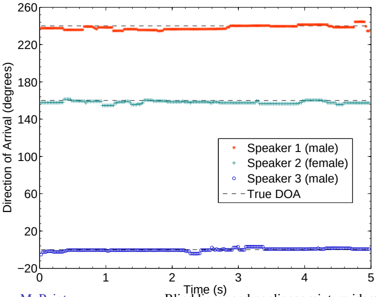
- Same simulated environment
- 4 intermittent sources
- Different history lengths (reponsiveness of the system)
- Different SNR conditions
- We measure a success rate of good source number estimation

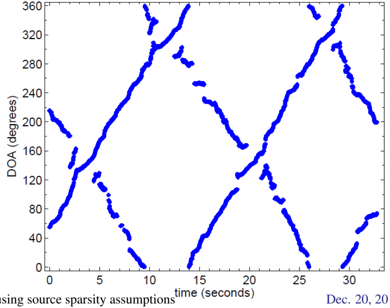
History length (s)	SNR (dB) 0	5	10	15	20
0.25s	44.1%	60.2%	77.6%	85.0%	88.4%
0.5s	61.2%	81.7%	94.2%	96.0%	96.6%
1s	82.1%	99.2%	100%	100.0%	100.0%

Performance (3)

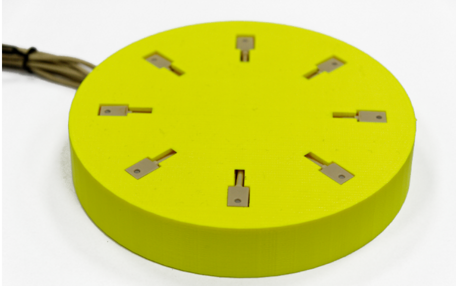
Experiments in real environment

- Real speakers, static or moving around the array, who speak continusouly
- Typical office room (same dimension as in simulations—4m $\times$ 6m $\times$ 3m) with A/C units (SNR$\simeq$15 dB)
- 8 omnidirectional Shure SM93 microphones, a TASCAM US2000 8-channel USB sound card, a Standard PC, Intel 3.00 GHz Core 2 CPU, 4 GB RAM, signal processing software in C++ and user interface in C♯
- Good trakking of the sources





Conclusion



- Real-time multiple source counting and localization method (55% of the available time)
- Source sparsity (including WDO assumption) was shown to help getting good performance (even if the acoustic model is not that realistic!)
- SSCM helps to get more accurate DOAs and to be much faster!
- Left available time may be used e.g. for speaker diarization or separation
- Effects of hardware (cheap microphones)?

Thank you for your attention

Questions?