Self-Calibration and Biconvex Compressive Sensing

Shuyang Ling

Department of Mathematics, UC Davis

July 12, 2017

Acknowledgements

Research in collaboration with:

Prof.Thomas Strohmer (UC Davis)

This work is sponsored by NSF-DMS and DARPA.

Outline

(a) Self-calibration and mathematical framework

(b) Biconvex compressive sensing in array signal processing

(c) SparseLift: a convex approach towards biconvex compressive sensing

(d) Theory and numerical simulations

Calibration

Calibration:

- Calibration is to adjust one device with the standard one.
- Why? To reduce or eliminate bias and inaccuracy.
- Difficult or even impossible to calibrate high-performance hardware.
- Self-calibration: Equip sensors with a smart algorithm which takes care of calibration automatically.

Calibration

Calibration:

- Calibration is to adjust one device with the standard one.
- Why? To reduce or eliminate bias and inaccuracy.
- Difficult or even impossible to calibrate high-performance hardware.
- Self-calibration: Equip sensors with a smart algorithm which takes care of calibration automatically.

Calibration realized by machine?

Uncalibrated device leads to imperfect sensing

We encounter imperfect sensing all the time: the sensing matrix $\boldsymbol{A}(\boldsymbol{h})$ depending on an unknown calibration parameter $\boldsymbol{h}$,

$$\boldsymbol{y} = \boldsymbol{A}(\boldsymbol{h})\boldsymbol{x} + \boldsymbol{w}.$$

This is too general to solve for $\boldsymbol{h}$ and $\boldsymbol{x}$ jointly.

Examples:

- Phase retrieval problem: $\boldsymbol{h}$ is the unknown phase of the Fourier transform of $\boldsymbol{x}$.
- Cryo-electron microscopy images: $\boldsymbol{h}$ can be the unknown orientation of a protein molecule and $\boldsymbol{x}$ is the particle.

Calibration realized by machine?

Uncalibrated device leads to imperfect sensing

We encounter imperfect sensing all the time: the sensing matrix $\boldsymbol{A}(\boldsymbol{h})$ depending on an unknown calibration parameter $\boldsymbol{h}$,

$$\boldsymbol{y} = \boldsymbol{A}(\boldsymbol{h})\boldsymbol{x} + \boldsymbol{w}.$$

This is too general to solve for $\boldsymbol{h}$ and $\boldsymbol{x}$ jointly.

Examples:

- Phase retrieval problem: $\boldsymbol{h}$ is the unknown phase of the Fourier transform of $\boldsymbol{x}$.
- Cryo-electron microscopy images: $\boldsymbol{h}$ can be the unknown orientation of a protein molecule and $\boldsymbol{x}$ is the particle.

A simplified but important model

Our focus:

One special case is to assume $\boldsymbol{A}(\boldsymbol{h})$ to be of the form

$$\boldsymbol{A}(\boldsymbol{h}) = \boldsymbol{D}(\boldsymbol{h})\boldsymbol{A}$$

where $\boldsymbol{D}(\boldsymbol{h})$ is an unknown diagonal matrix.

However, this seemingly simple model is very useful and mathematically nontrivial to analyze.

- Phase and gain calibration in array signal processing
- Blind deconvolution (image deblurring; joint channel and signal estimation, etc.)

A simplified but important model

Our focus:

One special case is to assume $\boldsymbol{A}(\boldsymbol{h})$ to be of the form

$$\boldsymbol{A}(\boldsymbol{h}) = \boldsymbol{D}(\boldsymbol{h})\boldsymbol{A}$$

where $\boldsymbol{D}(\boldsymbol{h})$ is an unknown diagonal matrix.

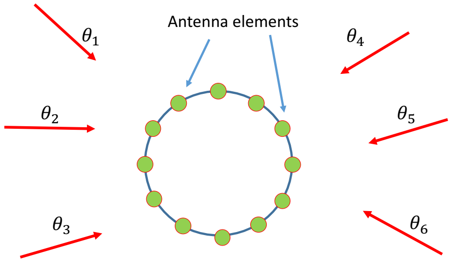
However, this seemingly simple model is very useful and mathematically nontrivial to analyze.

- Phase and gain calibration in array signal processing
- Blind deconvolution (image deblurring; joint channel and signal estimation, etc.)

Self-calibration in array signal processing

Calibration in the DOA (direction of arrival estimation)

One calibration issue comes from the unknown gains of the antennae caused by temperature or humidity.



Consider s signals impinging on an array of L antennae.

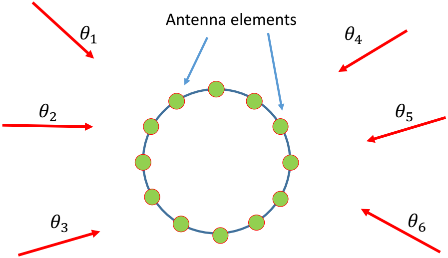
$$\boldsymbol{y} = \sum_{k=1}^{s} \boldsymbol{D}\boldsymbol{A}(\bar{\theta}_k)x_k + \boldsymbol{w}$$

where $\boldsymbol{D}$ is an unknown diagonal matrix and d_{ii} is the unknown gain for i-th sensor. $\boldsymbol{A}(\theta)$: array manifold. $\bar{\theta}_k$: unknown direction of arrival. $\{x_k\}_{k=1}^{s}$ are the impinging signals.

Self-calibration in array signal processing

Calibration in the DOA (direction of arrival estimation)

One calibration issue comes from the unknown gains of the antennae caused by temperature or humidity.



Consider s signals impinging on an array of L antennae.

$$\boldsymbol{y} = \sum_{k=1}^{s} \boldsymbol{D}\boldsymbol{A}(\bar{\theta}_k)x_k + \boldsymbol{w}$$

where $\boldsymbol{D}$ is an unknown diagonal matrix and d_{ii} is the unknown gain for i-th sensor. $\boldsymbol{A}(\theta)$: array manifold. $\bar{\theta}_k$: unknown direction of arrival. $\{x_k\}_{k=1}^{s}$ are the impinging signals.

How is it related to compressive sensing?

Discretize the manifold function $\boldsymbol{A}(\theta)$ over $[-\pi \leq \theta < \pi]$ on N grid points.

$$\boldsymbol{y} = \boldsymbol{D}\boldsymbol{A}\boldsymbol{x} + \boldsymbol{w}$$

where

$$\boldsymbol{A} = \begin{bmatrix} | & \cdots & | \\ \boldsymbol{A}(\theta_1) & \cdots & \boldsymbol{A}(\theta_N) \\ | & \cdots & | \end{bmatrix} \in \mathbb{C}^{L\times N}$$

- To achieve high resolution, we usually have $L \leq N$.
- $\boldsymbol{x} \in \mathbb{C}^{N\times 1}$ is s-sparse. Its s nonzero entries correspond to the directions of signals. Moreover, we don't know the locations of nonzero entries.
- Subspace constraint: assume $\boldsymbol{D} = \mathrm{diag}(\boldsymbol{B}\boldsymbol{h})$ where $\boldsymbol{B}$ is a known $L \times K$ matrix and $K < L$.
- Number of constraints: L; number of unknowns: $K + s$.

How is it related to compressive sensing?

Discretize the manifold function $\boldsymbol{A}(\theta)$ over $[-\pi \leq \theta < \pi]$ on N grid points.

$$\boldsymbol{y} = \boldsymbol{D}\boldsymbol{A}\boldsymbol{x} + \boldsymbol{w}$$

where

$$\boldsymbol{A} = \begin{bmatrix} | & \cdots & | \\ \boldsymbol{A}(\theta_1) & \cdots & \boldsymbol{A}(\theta_N) \\ | & \cdots & | \end{bmatrix} \in \mathbb{C}^{L\times N}$$

- To achieve high resolution, we usually have $L \leq N$.
- $\boldsymbol{x} \in \mathbb{C}^{N\times 1}$ is s-sparse. Its s nonzero entries correspond to the directions of signals. Moreover, we don't know the locations of nonzero entries.
- Subspace constraint: assume $\boldsymbol{D} = \text{diag}(\boldsymbol{B}\boldsymbol{h})$ where $\boldsymbol{B}$ is a known $L \times K$ matrix and $K < L$.
- Number of constraints: L; number of unknowns: $K + s$.

Self-calibration and biconvex compressive sensing

Goal: Find $(\boldsymbol{h}, \boldsymbol{x})$ s.t. $\boldsymbol{y} = \text{diag}(\boldsymbol{Bh})\boldsymbol{Ax} + \boldsymbol{w}$ and $\boldsymbol{x}$ is sparse.

Biconvex compressive sensing

We are solving a biconvex (not convex) optimization problem to recover sparse signal $\boldsymbol{x}$ and calibrating parameter $\boldsymbol{h}$.

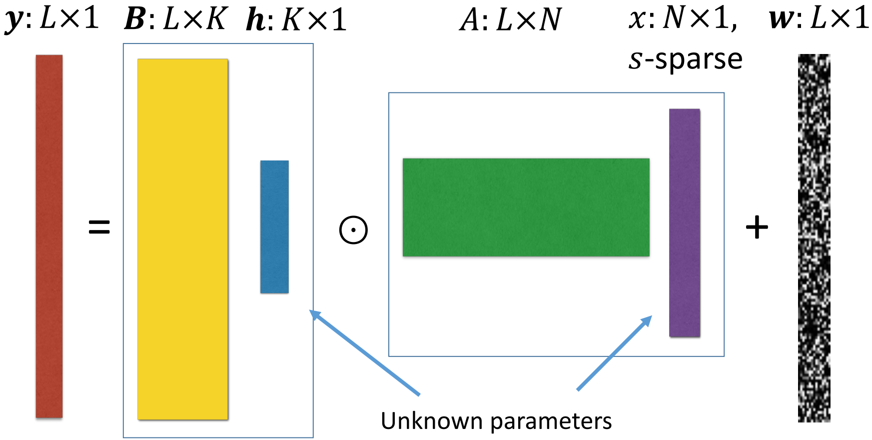
$$\min_{\boldsymbol{h},\boldsymbol{x}} \|\text{diag}(\boldsymbol{Bh})\boldsymbol{Ax} - \boldsymbol{y}\|^2 + \lambda\|\boldsymbol{x}\|_1$$

$\boldsymbol{A} \in \mathbb{C}^{L\times N}$ and $\boldsymbol{B} \in \mathbb{C}^{L\times K}$ are known. $\boldsymbol{h} \in \mathbb{C}^{K\times 1}$ and $\boldsymbol{x} \in \mathbb{C}^{N\times 1}$ are unknown. $\boldsymbol{x}$ is sparse.

Remark: If $\boldsymbol{h}$ is known, $\boldsymbol{x}$ can be recovered; if $\boldsymbol{x}$ is known, we can find $\boldsymbol{h}$ as well. Regarding identifiability issue, See [Lee, etc. 15].

Biconvex compressive sensing

Goal: we want to find $\boldsymbol{h}$ and a sparse $\boldsymbol{x}$ from $\boldsymbol{y}$, $\boldsymbol{B}$ and $\boldsymbol{A}$.



Convex approach and lifting

Two-step convex approach

(a) Lifting: convert bilinear to linear constraints

- Widely used in phase retrieval [Candès, etc, 11], blind deconvolution [Ahmed, etc, 11], etc...

(b) Solving a convex relaxation (semi-definite program) to recover $\boldsymbol{h}_0\boldsymbol{x}_0^*$.

Lifting: from bilinear to linear

Step 1: lifting

Let $\boldsymbol{a}_i$ be the i-th column of $\boldsymbol{A}^*$ and $\boldsymbol{b}_i$ be the i-th column of $\boldsymbol{B}^*$.

$$y_i = (\boldsymbol{B}\boldsymbol{h}_0)_i \boldsymbol{x}_0^* \boldsymbol{a}_i + w_i = \boldsymbol{b}_i^* \boldsymbol{h}_0 \boldsymbol{x}_0^* \boldsymbol{a}_i + w_i.$$

Let $\boldsymbol{X}_0 := \boldsymbol{h}_0 \boldsymbol{x}_0^*$ and define the linear operator $\mathcal{A} : \mathbb{C}^{K\times N} \to \mathbb{C}^L$ as,

$$\mathcal{A}(\boldsymbol{Z}) := \{\boldsymbol{b}_i^* \boldsymbol{Z} \boldsymbol{a}_i\}_{i=1}^L = \{\langle \boldsymbol{Z}, \boldsymbol{b}_i \boldsymbol{a}_i^* \rangle\}_{i=1}^L.$$

Then, there holds

$$\boldsymbol{y} = \mathcal{A}(\boldsymbol{X}_0) + \boldsymbol{w}.$$

In this way, $\mathcal{A}^*(\boldsymbol{z}) = \sum_{i=1}^L z_i \boldsymbol{b}_i \boldsymbol{a}_i^* : \mathbb{C}^L \to \mathbb{C}^{K\times N}$.

Rank-1 matrix recovery

Lifting: recovery of a rank - 1 and row-sparse matrix

$$\text{Find } \boldsymbol{Z} \quad \text{s.t.} \quad \begin{aligned} &\text{rank}(\boldsymbol{Z}) = 1 \\ &\mathcal{A}(\boldsymbol{Z}) = \mathcal{A}(\boldsymbol{X}_0) \\ &\boldsymbol{Z} \text{ has sparse rows} \end{aligned}$$

- $\|\boldsymbol{X}_0\|_0 = Ks$ where $\boldsymbol{X}_0 = \boldsymbol{h}_0\boldsymbol{x}_0^*$, $\boldsymbol{h}_0 \in \mathbb{C}^K$ and $\boldsymbol{x}_0 \in \mathbb{C}^N$ with $\|\boldsymbol{x}_0\|_0 = s$.

$$\boldsymbol{Z} = \begin{bmatrix} 0 & 0 & h_1 x_{i_1} & 0 & \cdots & 0 & h_1 x_{i_s} & 0 & \cdots & 0 \\ 0 & 0 & h_2 x_{i_1} & 0 & \cdots & 0 & h_2 x_{i_s} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & h_K x_{i_1} & 0 & \cdots & 0 & h_K x_{i_s} & 0 & \cdots & 0 \end{bmatrix}_{K \times N}$$

- An NP-hard problem to find such a rank-1 and sparse matrix.

Rank-1 matrix recovery

Lifting: recovery of a rank - 1 and row-sparse matrix

$$\text{Find } \boldsymbol{Z} \quad \text{s.t.} \quad \begin{aligned} &\operatorname{rank}(\boldsymbol{Z}) = 1 \\ &\mathcal{A}(\boldsymbol{Z}) = \mathcal{A}(\boldsymbol{X}_0) \\ &\boldsymbol{Z} \text{ has sparse rows} \end{aligned}$$

- $\|\boldsymbol{X}_0\|_0 = Ks$ where $\boldsymbol{X}_0 = \boldsymbol{h}_0\boldsymbol{x}_0^*$, $\boldsymbol{h}_0 \in \mathbb{C}^K$ and $\boldsymbol{x}_0 \in \mathbb{C}^N$ with $\|\boldsymbol{x}_0\|_0 = s$.

$$\boldsymbol{Z} = \begin{bmatrix} 0 & 0 & h_1 x_{i_1} & 0 & \cdots & 0 & h_1 x_{i_s} & 0 & \cdots & 0 \\ 0 & 0 & h_2 x_{i_1} & 0 & \cdots & 0 & h_2 x_{i_s} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & h_K x_{i_1} & 0 & \cdots & 0 & h_K x_{i_s} & 0 & \cdots & 0 \end{bmatrix}_{K\times N}$$

- An NP-hard problem to find such a rank-1 and sparse matrix.

SparseLift

$\|\boldsymbol{Z}\|_*$: nuclear norm and $\|\boldsymbol{Z}\|_1$: ℓ_1-norm of vectorized $\boldsymbol{Z}$.

A popular way: nuclear norm + ℓ_1- minimization

$$\min \|\boldsymbol{Z}\|_1 + \lambda\|\boldsymbol{Z}\|_* \quad \text{s.t.} \quad \mathcal{A}(\boldsymbol{Z}) = \mathcal{A}(\boldsymbol{X}_0), \quad \lambda \geq 0.$$

However, combination of multiple norms may not do any better. [Oymak, etc. 12].

SparseLift

$$\min \|\boldsymbol{Z}\|_1 \quad \text{s.t.} \quad \mathcal{A}(\boldsymbol{Z}) = \mathcal{A}(\boldsymbol{X}_0).$$

Idea: Lift the recovery problem of two unknown vectors to a matrix-valued problem and exploit sparsity through ℓ_1-minimization.

SparseLift

$\|\boldsymbol{Z}\|_*$: nuclear norm and $\|\boldsymbol{Z}\|_1$: ℓ_1-norm of vectorized $\boldsymbol{Z}$.

A popular way: nuclear norm + ℓ_1- minimization

$$\min \|\boldsymbol{Z}\|_1 + \lambda \|\boldsymbol{Z}\|_* \quad \text{s.t.} \quad \mathcal{A}(\boldsymbol{Z}) = \mathcal{A}(\boldsymbol{X}_0), \quad \lambda \geq 0.$$

However, combination of multiple norms may not do any better. [Oymak, etc. 12].

SparseLift

$$\min \|\boldsymbol{Z}\|_1 \quad \text{s.t.} \quad \mathcal{A}(\boldsymbol{Z}) = \mathcal{A}(\boldsymbol{X}_0).$$

Idea: Lift the recovery problem of two unknown vectors to a matrix-valued problem and exploit sparsity through ℓ_1-minimization.

Main theorem

Theorem: [Ling-Strohmer, 2015]

Recall the model:

$$\boldsymbol{y} = \boldsymbol{D}\boldsymbol{A}\boldsymbol{x}, \quad \boldsymbol{D} = \operatorname{diag}(\boldsymbol{B}\boldsymbol{h}),$$

where

(a) $\boldsymbol{B}$ is an $L \times K$ DFT tall matrix with $\boldsymbol{B}^*\boldsymbol{B} = \boldsymbol{I}_K$

(b) $\boldsymbol{A}$ is an $L \times N$ real Gaussian random matrix or a random Fourier matrix.

Then SparseLift recovers $\boldsymbol{X}_0$ exactly with high probability if

$$L = \mathcal{O}(\underbrace{K}_{\text{dimension of } \boldsymbol{h}} \quad \underbrace{s}_{\text{level of sparsity}} \log^2 L)$$

where $Ks = \|\boldsymbol{X}_0\|_0$.

Comments

- It is shown that $\ell_{1\text{-}2}$ minimization also works (exploit group sparsity) [Flinth, 17].
- $\min \|\boldsymbol{X}\|_*$ fails if $L < N$.

$\min \|\boldsymbol{X}\|_*$	$L = \mathcal{O}(K + N)$
$\min \|\boldsymbol{X}\|_1$	$L = \mathcal{O}(\mathbf{Ks} \log KN)$

- Solving ℓ_1-minimization is easier and cheaper than solving SDP.
- Compared with Compressive Sensing

Compressive Sensing	$L = \mathcal{O}(\mathbf{s} \log N)$
Our Case	$L = \mathcal{O}(K\mathbf{s} \log KN)$

- Believed to be optimal if one uses the 'Lifting' technique. It is unknown whether any algorithm would work for $L = \mathcal{O}(K + s)$.

$\min\|\cdot\|_1 + \lambda\|\cdot\|_*$ does not do any better than $\min\|\cdot\|_1$.
White: Success, Black: Failure

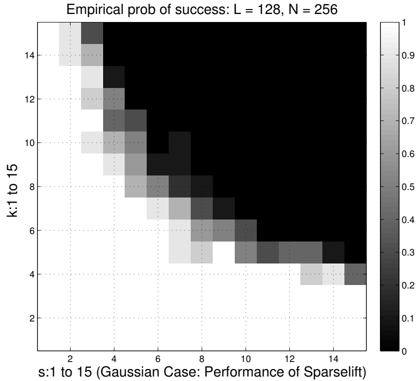


Figure: SparseLift

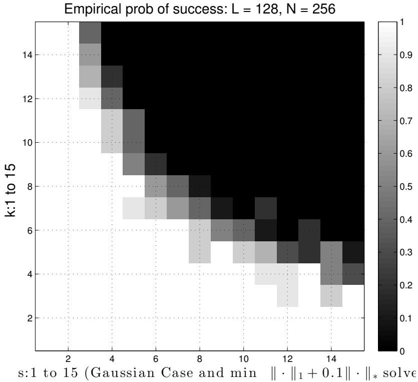


Figure: $\min\|\cdot\|_1 + 0.1\|\cdot\|_*$

$L = 128, N = 256$. $\boldsymbol{A}$: Gaussian and $\boldsymbol{B}$: Non-random partial Fourier matrix. 10 experiments for each pair (K, s), $1 \le K, s \le 15$.

Minimal L is nearly proportional to Ks

L : 10 to 400; $N = 512$; $\boldsymbol{A}$: Gaussian random matrices;
$\boldsymbol{B}$: first K columns of a DFT matrix.

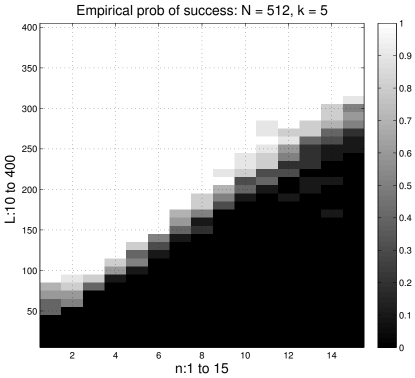


Figure: Fix $K = 5$

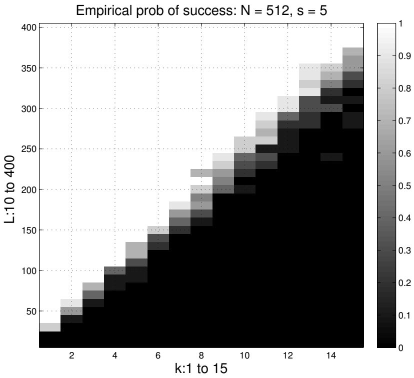


Figure: Fix $s = 5$

Stability theory

Assume that $\boldsymbol{y}$ is contaminated by noise, namely, $\boldsymbol{y} = \mathcal{A}(\boldsymbol{X}_0) + \boldsymbol{w}$ with $\|\boldsymbol{w}\| \leq \eta$, we solve the following program to recover $\boldsymbol{X}_0$,

$$\min \|\boldsymbol{Z}\|_1 \quad \text{s.t.} \quad \|\mathcal{A}(\boldsymbol{Z}) - \boldsymbol{y}\| \leq \eta.$$

Theorem

If $\boldsymbol{A}$ is either a Gaussian random matrix or a random Fourier matrix,

$$\|\hat{\boldsymbol{X}} - \boldsymbol{X}_0\|_F \leq (C_0 + C_1\sqrt{Ks})\eta$$

with high probability. L satisfies the condition in the noiseless case. Both C_0 and C_1 are constants.

Stability theory

Assume that $\boldsymbol{y}$ is contaminated by noise, namely, $\boldsymbol{y} = \mathcal{A}(\boldsymbol{X}_0) + \boldsymbol{w}$ with $\|\boldsymbol{w}\| \leq \eta$, we solve the following program to recover $\boldsymbol{X}_0$,

$$\min \|\boldsymbol{Z}\|_1 \quad \text{s.t. } \|\mathcal{A}(\boldsymbol{Z}) - \boldsymbol{y}\| \leq \eta.$$

Theorem

If $\boldsymbol{A}$ is either a Gaussian random matrix or a random Fourier matrix,

$$\|\hat{\boldsymbol{X}} - \boldsymbol{X}_0\|_F \leq (C_0 + C_1\sqrt{Ks})\eta$$

with high probability. L satisfies the condition in the noiseless case. Both C_0 and C_1 are constants.

Numerical example: relative error vs SNR

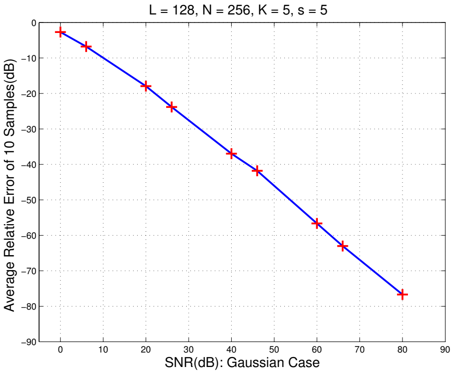


Figure: $\boldsymbol{A}$: Gaussian matrix

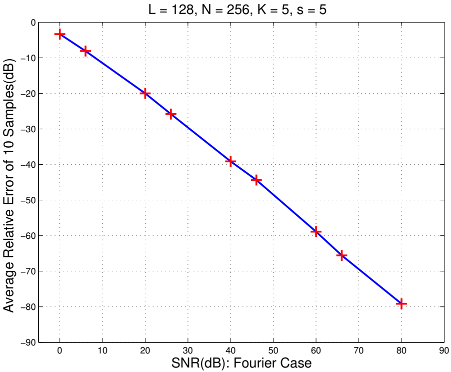
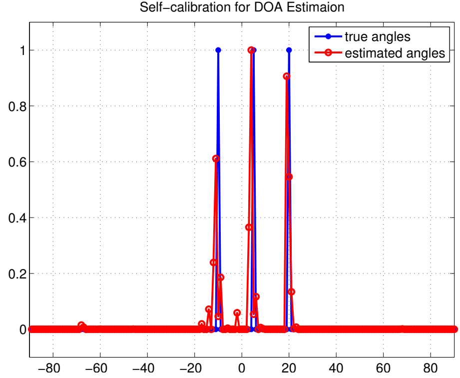


Figure: $\boldsymbol{A}$: random Fourier matrix

Remarks: $L = 128, N = 256, K = s = 5.$

DOA with a single snapshot

Assume we have $L = 64$ sensors on one circle (circular array) and the gain $\boldsymbol{d} = \boldsymbol{B}\boldsymbol{h}$ where $\boldsymbol{B} \in \mathbb{C}^{64\times 4}$ and $\boldsymbol{h}$ is complex Gaussian. The discretization of the angle consists of $N = 180$ grid points over $[-89^o, 90^o]$ and SNR is 25dB.



Outlook and Conclusion

Conclusion:

- Is it possible to recover $(\boldsymbol{h}, \boldsymbol{x})$ with $L = \mathcal{O}(K + s)$ measurements?
- Consider multiple snapshots instead of one single snapshots.
- **See details**:
  1. Self-calibration and biconvex compressive sensing. *Inverse Problems* 31 (11), 115002
  2. Blind deconvolution meets blind demixing: algorithms and performance bounds, *IEEE Transactions on Information Theory* 63 (7), 4497-4520
  3. Rapid, robust, and reliable blind deconvolution via nonconvex optimization, *arXiv:1606.04933*.
  4. Regularized gradient descent: a nonconvex recipe for fast joint blind deconvolution and demixing *arXiv:1703.08642*.