Greedy algorithms for multi-channel sparse recovery

Public ISP seminar at UCL

Jean-François Determe

Summary

- **Main topic:** How noise impacts (S)OMP & overview of noise stabilization with SOMP
- **Outline**:
  - Introduction compressive sensing (7 slides -> 9 minutes)
  - Support recovery algorithms (3 slides -> 5 minutes)
  - Multiple measurement vector signal models (4 slides -> 6 minutes)

  - Analysis of SOMP with noise (8 slides -> 12 minutes)
  - SOMP with noise stabilization (6 slides -> 8 minutes)
  - Conclusion (2 minutes)
- **Total time for presentation**: about 40-45 minutes + Q&A
- Presentation = only an overview of my work (no technical details, not every contribution)

Outline

- Introduction to compressive sensing
- Support recovery algorithms
- Multiple measurement vector signal models
- Analysis of SOMP with noise
- SOMP with noise stabilization
- Conclusion

Outline

- **Introduction to compressive sensing**
- Support recovery algorithms
- Multiple measurement vector signal models
- Analysis of SOMP with noise
- SOMP with noise stabilization
- Conclusion

Compressive sensing (CS)

Idea : Observe and recover a signal $\boldsymbol{f} \in \mathbb{R}^n$ using $m \ll n$ linear measurements:

$$\boldsymbol{y} = \boldsymbol{\Phi} \boldsymbol{f} \in \mathbb{R}^m \quad \text{where} \quad \boldsymbol{\Phi} \in \mathbb{R}^{m \times n} \text{ describes the measurement process}$$

Problem : Since $m \ll n$, arbitrary signals $\boldsymbol{f}$ cannot be recovered
Solution : Assume prior knowledge/structure about $\boldsymbol{f}$

Sparsity : $\boldsymbol{f}$ can be expressed using $s < m$ vectors from the appropriate o.n. basis $\boldsymbol{\Psi}$

$$\boldsymbol{f} = \boldsymbol{\Psi} \boldsymbol{x} = \sum_{j=1}^{n} x_j \psi_j$$

Few non-zero x_j

Support explanation

$$\boldsymbol{f} = \boldsymbol{\Psi x} = \sum_{j=1}^{n} x_j \boldsymbol{\psi}_j = \sum_{j \in \mathrm{supp}(\boldsymbol{x})} x_j \boldsymbol{\psi}_j \qquad \mathrm{supp}(\boldsymbol{x}) := \{j : x_j \neq 0\}$$

$$\boldsymbol{y} = \boldsymbol{\Psi x} = \begin{bmatrix} \boldsymbol{\psi}_1 & \boldsymbol{\psi}_2 & \boldsymbol{\psi}_3 & \boldsymbol{\psi}_4 & \boldsymbol{\psi}_5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = x_1 \boldsymbol{\psi}_1 + x_2 \boldsymbol{\psi}_2 + x_3 \boldsymbol{\psi}_3 + \ldots$$

- Sparsity example:

$$\boldsymbol{y} = \boldsymbol{\Psi x} = \begin{bmatrix} \boldsymbol{\psi}_1 & \boldsymbol{\psi}_2 & \boldsymbol{\psi}_3 & \boldsymbol{\psi}_4 & \boldsymbol{\psi}_5 \end{bmatrix} \begin{bmatrix} 0 \\ x_2 \\ 0 \\ 0 \\ x_5 \end{bmatrix} \qquad \mathrm{supp}(\boldsymbol{x}) = \{2, 5\}$$

$$= x_2 \boldsymbol{\psi}_2 + x_5 \boldsymbol{\psi}_5$$

Compressive sensing (CS)

Idea : Observe and recover a signal $\boldsymbol{f} \in \mathbb{R}^n$ using $m \ll n$ linear measurements:

$$\boldsymbol{y} = \boldsymbol{\Phi} \boldsymbol{f} \in \mathbb{R}^m \quad \text{where} \quad \boldsymbol{\Phi} \in \mathbb{R}^{m \times n} \text{ describes the measurement process}$$

Problem : Since $m \ll n$, arbitrary signals $\boldsymbol{f}$ cannot be recovered

Solution : Assume prior knowledge/structure about $\boldsymbol{f}$

Sparsity : $\boldsymbol{f}$ can be expressed using $s < m$ vectors from the appropriate o.n. basis $\boldsymbol{\Psi}$

$$\boldsymbol{f} = \boldsymbol{\Psi} \boldsymbol{x} = \sum_{j=1}^{n} x_j \psi_j = \sum_{j \in \text{supp}(\boldsymbol{x})} x_j \psi_j \longrightarrow \boxed{\boldsymbol{y} = \boldsymbol{\Phi} \boldsymbol{\Psi} \boldsymbol{x}}$$

Few non-zero x_j | $\text{supp}(\boldsymbol{x}) := \{j : x_j \neq 0\}$ has low cardinality

In practice : $\boldsymbol{\Phi}$ is generated randomly using sub-Gaussian distributions

⟶ $\boldsymbol{\Phi}$ and $\boldsymbol{\Phi}\boldsymbol{\Psi}$ satisfy the required properties for CS with similar probabilities

⟶ Simplification: $\boxed{\boldsymbol{\Psi} = \boldsymbol{I}_{n \times n} \text{ and } \boldsymbol{y} = \boldsymbol{\Phi} \boldsymbol{x}} = \sum_{j=1}^{n} x_j \phi_j = \sum_{j \in \text{supp}(\boldsymbol{x})} x_j \phi_j$

Compressive sensing (CS)

→ Simplification: $\boldsymbol{\Psi} = \boldsymbol{I}_{n \times n}$ and $\boxed{\boldsymbol{y} = \boldsymbol{\Phi x}} = \sum_{j \in \mathrm{supp}(\boldsymbol{x})} x_j \boldsymbol{\phi}_j$

In practice: Meas. matrix $\boldsymbol{\Phi}$ can have Gaussian entries or Rademacher entries (+/- 1 with equal probabilities) + normalization factor

Two ways to understand why random projections are neat

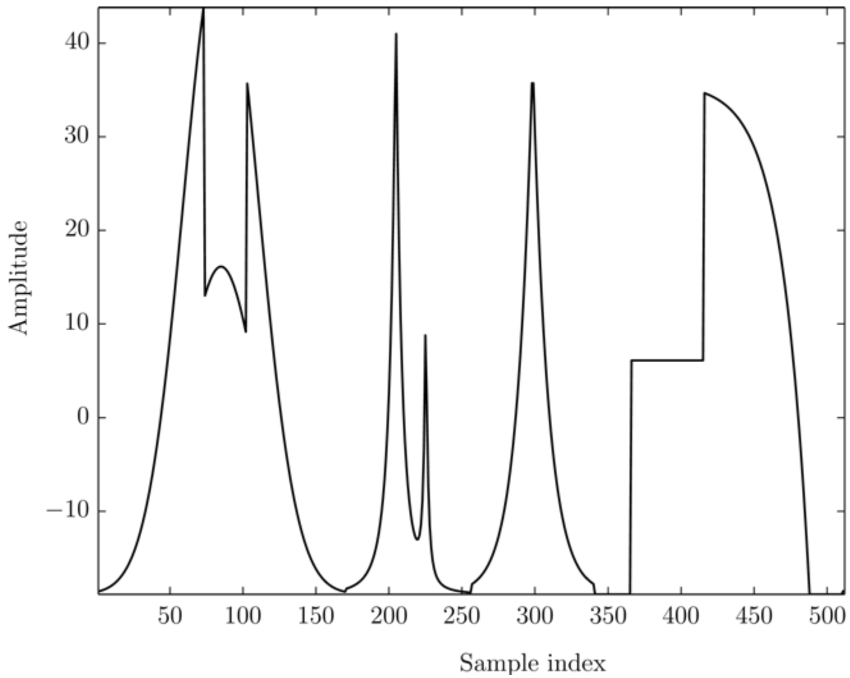
- Recovering $\boldsymbol{x}$ is more easy using random, diverse projections (very similar projections are not efficient to capture information about $\boldsymbol{x}$)
- Proper random entries in $\boldsymbol{\Phi}$ make the atoms $\boldsymbol{\phi}_j$ ``more orthogonal'' to one another. Easier to distinguish atoms in the sum

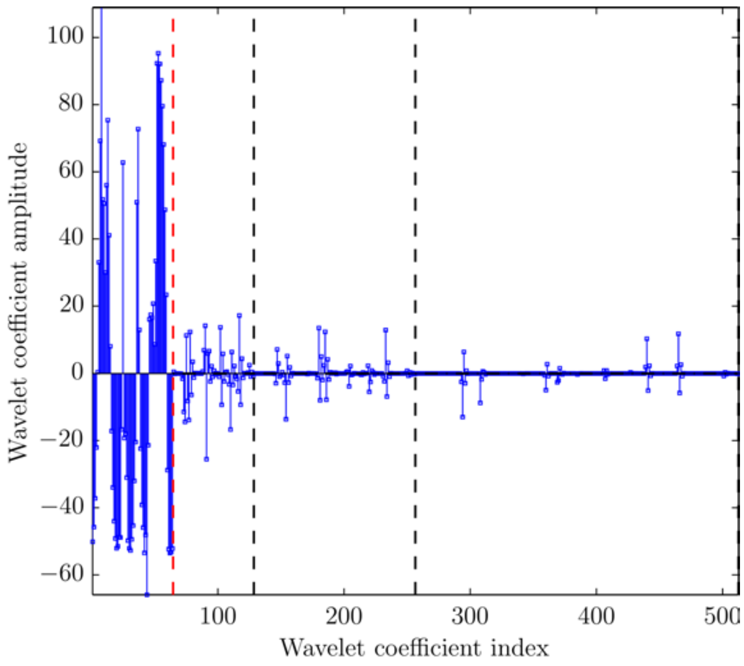
$$\boldsymbol{y} = \sum_{j \in \mathrm{supp}(\boldsymbol{x})} x_j \boldsymbol{\phi}_j$$

Quantity $\langle \boldsymbol{y}, \boldsymbol{\phi}_j \rangle$ becomes a good proxy for x_j

Sparse (compressible) 1D signal Example

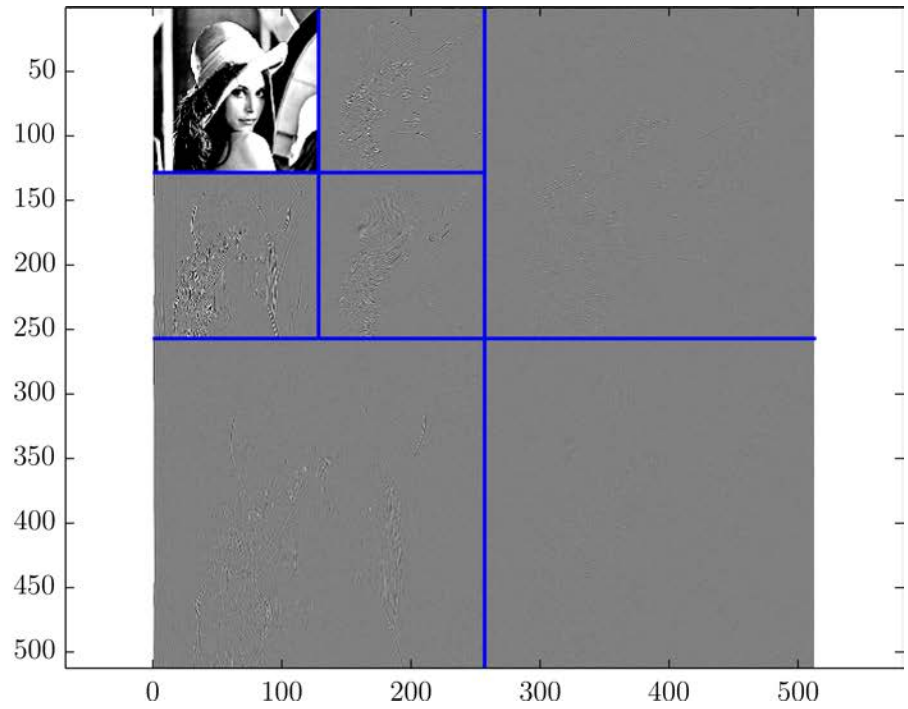
- D14 wavelets – Level of decomposition = 3





Sparse (compressible) 2D signal Example

- D8 wavelets – Level of decomposition = 2



RIP and RICs

Idea: Observe and recover a sparse signal $\boldsymbol{x} \in \mathbb{R}^n$ using $m << n$ linear measurements:

$$\boldsymbol{y} = \boldsymbol{\Phi x} \in \mathbb{R}^m \quad \text{where} \quad \boldsymbol{\Phi} \in \mathbb{R}^{m \times n} \text{ describes the measurement process}$$

Question: How to quantify how good the measurement matrix $\boldsymbol{\Phi}$ is?
Solution: Restricted isometry property (and associated RICs)

RIP: $\boldsymbol{\Phi}$ satisfies the RIP (with a RIC of order s δ_s) if

$$(1 - \delta_s)\|\boldsymbol{u}\|_2^2 \leq \|\boldsymbol{\Phi u}\|_2^2 \leq (1 + \delta_s)\|\boldsymbol{u}\|_2^2$$

for any s-sparse vector $\boldsymbol{u}$

Interpretation: RICs quantify to what extent a measurement matrix is suitable for CS

Good RIC	Bad RIC
$\delta_s \simeq 0$	$\delta_s \simeq 1$

Outline

- Introduction to compressive sensing
- **Support recovery algorithms**
- Multiple measurement vector signal models
- Analysis of SOMP with noise
- SOMP with noise stabilization
- Conclusion

Support recovery algorithms

Idea: Observe and recover a sparse signal $\boldsymbol{x} \in \mathbb{R}^n$ using $m << n$ linear measurements:

$$\boldsymbol{y} = \boldsymbol{\Phi x} \in \mathbb{R}^m \quad \text{where} \quad \boldsymbol{\Phi} \in \mathbb{R}^{m \times n} \text{ describes the measurement process}$$

Several algorithms can recover the support of $\boldsymbol{x}$ on the basis of $\boldsymbol{\Phi}$ and $\boldsymbol{y}$

Two main classes of support recovery algorithms

- Algorithms based upon **convex optimization** (e.g., basis pursuit, basis pursuit denoising, and Dantzig selector)
  - Higher computational requirements (CPU time + memory)
  - Best performance (theoretical + numerical)
- **Greedy** algorithms (e.g., MP, OMP, CoSaMP, and SP)
  - Lower computational requirements
  - May be less reliable than, e.g., basis pursuit.

My thesis focuses on greedy algorithms (OMP-like algorithms)

Orthogonal matching pursuit

$$\boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{x} = \sum_{j \in \mathcal{S}} x_j \boldsymbol{\phi}_j$$

Orthogonal matching pursuit (OMP) tries to express the measurement vector $\boldsymbol{y}$ using columns from $\boldsymbol{\Phi}$

➡ **Generates an estimated support** (its size is prescribed beforehand)

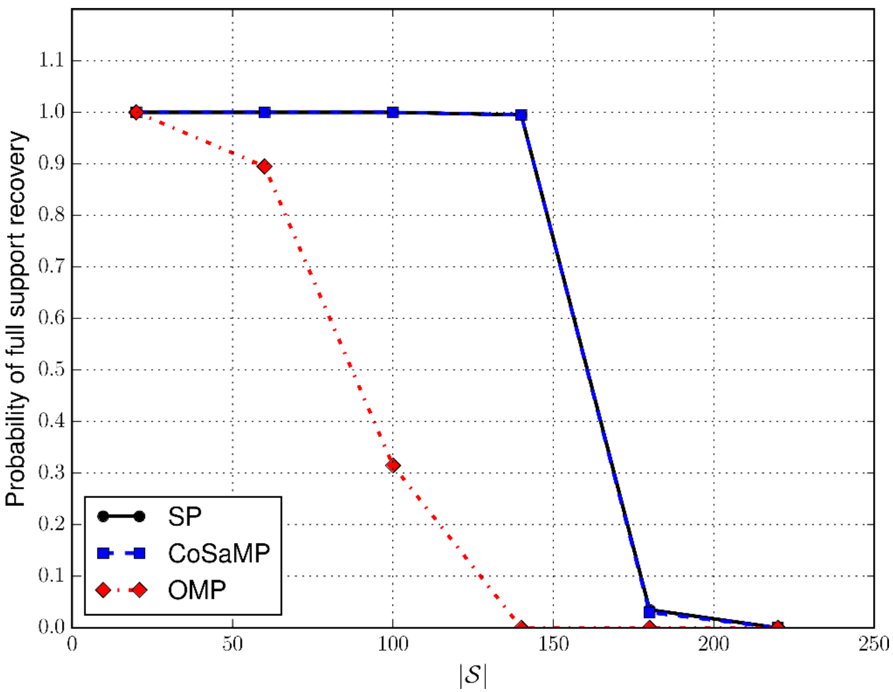
- OMP = iterative algorithm
  - Adds **one** element to estimated support at each iteration
- At each iteration:
  - Look for atom/column ϕ_j most closely resembling the measurement vector $\boldsymbol{y}$ -> inner product
  - Add this atom to estimated support
  - Remove the atom contribution to the measurements (-> approximation only)
    If $\mathcal{S}_t$ = estimated support at iteration *t* => build proxy for

$$\boldsymbol{\Phi}_{\mathcal{S}\setminus\mathcal{S}_t} \boldsymbol{x}_{\mathcal{S}\setminus\mathcal{S}_t} = \sum_{j \in \mathcal{S}\setminus\mathcal{S}_t} x_j \boldsymbol{\phi}_j$$

Performance comparison for greedy algorithms

Noiseless case

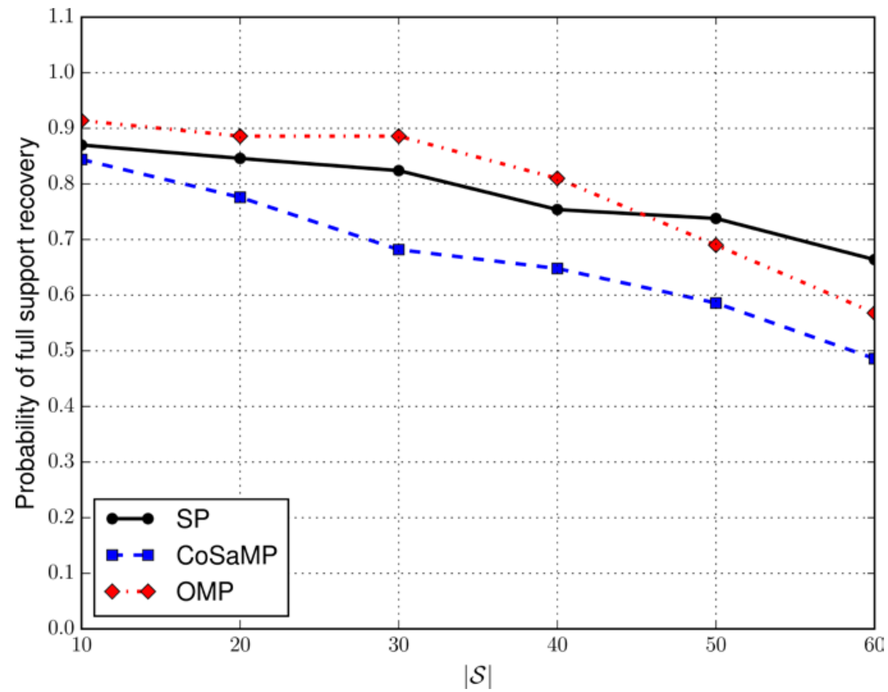
$$\boldsymbol{y} = \boldsymbol{\Phi x}$$



OMP **not** competitive

Noisy case

$$\boldsymbol{y} = \boldsymbol{\Phi x} + \boldsymbol{e}$$

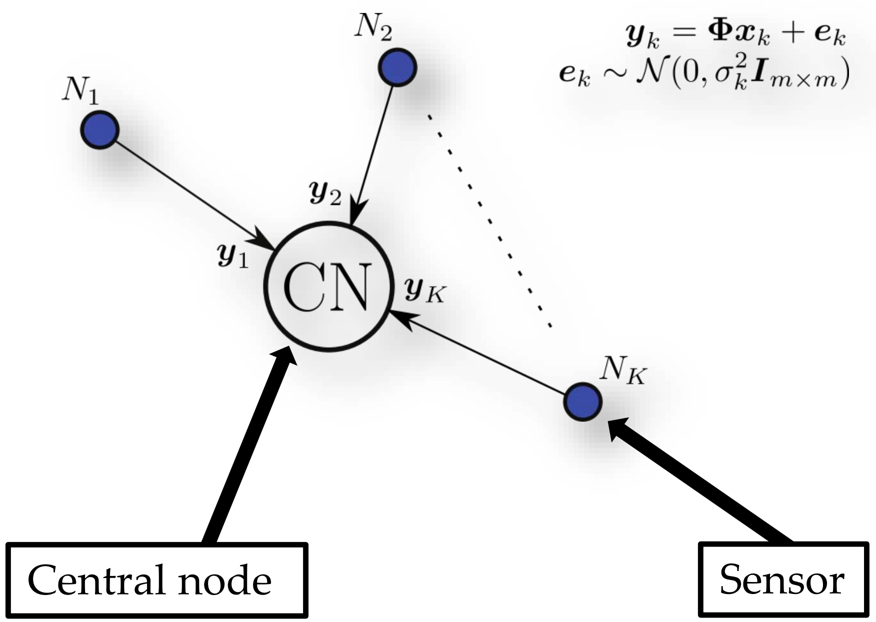


OMP competitive

Outline

- Introduction to compressive sensing
- Support recovery algorithms
- **Multiple measurement vector signal models**
- Analysis of SOMP with noise
- SOMP with noise stabilization
- Conclusion

MMV Signal model (1)



- K sensors observe **a common physical phenomenon** (chemical composition, image, wireless spectrum, etc.)
- **Local variability** ➡ unequal x_k
- Yet: **observed rough structure is identical** for each sensor

➡ $\text{supp}(x_k)$ are similar (if not equal)

<u>Joint support</u>: $\mathcal{S} := \bigcup_{1 \le k \le K} \text{supp}(x_k)$

- Sensors with different noise variances

<u>Extension of basic CS model</u> : - Multiple measurement vector (MMV) signal model
- Additive Gaussian noise with $\neq$ variances

K sparse signals/measurement channels/measurement vectors:

$$y_k = \Phi x_k + e_k \quad (1 \le k \le K) \quad \text{where} \quad e_k \sim \mathcal{N}(0, \sigma_k^2 I_{m \times m})$$

MMV Signal model (2)

Extension of basic CS model : - Multiple measurement vector (MMV) signal model
- Additive Gaussian noise with $\neq$ variances

K sparse signals/measurement channels/measurement vectors:

$$\boldsymbol{y}_k = \boldsymbol{\Phi}\boldsymbol{x}_k + \boldsymbol{e}_k \quad (1 \leq k \leq K) \quad \text{where} \quad \boldsymbol{e}_k \sim \mathcal{N}(0, \sigma_k^2 \boldsymbol{I}_{m \times m})$$

With matrices: $\boldsymbol{Y} = \boldsymbol{\Phi}\boldsymbol{X} + \boldsymbol{E} \longrightarrow \boldsymbol{X} \in \mathbb{R}^{n \times K}$, $\boldsymbol{\Phi} \in \mathbb{R}^{m \times n}$, and $\boldsymbol{Y}, \boldsymbol{E} \in \mathbb{R}^{m \times K}$

$\boldsymbol{Y} = (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_K)$ $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_K)$ $\boldsymbol{E} = (\boldsymbol{e}_1, \ldots, \boldsymbol{e}_K)$

Joint support: $\mathcal{S} := \text{supp}(\boldsymbol{X}) := \bigcup_{1 \leq k \leq K} \text{supp}(\boldsymbol{x}_k)$

- Number of measurements = m
- Number of atoms/columns = n
- Number of measurement vectors/ channels = K
- $\boldsymbol{\Phi} \in \mathbb{R}^{m \times n}$

Objective: Recover the joint support on the basis of $\{\boldsymbol{y}_k\}_{1 \leq k \leq K}$ and $\boldsymbol{\Phi}$.

Remarks on MMV signal models

$$\boldsymbol{y}_k = \boldsymbol{\Phi}\boldsymbol{x}_k + \boldsymbol{e}_k \;\; (1 \leq k \leq K) \;\; \text{where} \;\; \boldsymbol{e}_k \sim \mathcal{N}(0, \sigma_k^2 \boldsymbol{I}_{m \times m})$$

- MMV extensions of SMV algorithms are available (e.g., SOMP, SCoSaMP, SSP)
- Focus of this presentation = SOMP **exclusively**
- Several applications for MMV signal models:
  - <u>Source localization:</u> each measurement vector corresponds to a specific time instant
  - <u>Localization in 5G networks</u>
  - <u>Spectrum sensing/sub-Nyquist acquisition</u> with the modulated wideband converter

Simultaneous orthogonal matching pursuit

$$\boldsymbol{y}_k = \boldsymbol{\Phi}\boldsymbol{x}_k + \boldsymbol{e}_k \quad \text{where } \boldsymbol{e}_k \sim \mathcal{N}(\boldsymbol{0}, \sigma_k^2 \boldsymbol{I}) \text{ and } \boldsymbol{\Phi}\boldsymbol{x}_k = \sum_{j \in \mathcal{S}} (\boldsymbol{x}_k)_j \boldsymbol{\phi}_j$$

Simultaneous orthogonal matching pursuit (SOMP) tries to jointly express the K measurement vectors $\boldsymbol{y}_k$ using a **unique** set of columns from $\boldsymbol{\Phi}$

➡ **Joint support recovery**, *i.e.*, one common support for all the sparse signals $\boldsymbol{x}_k$

- SOMP = iterative algorithm
  - Adds **one** element to estimated support at each iteration
- At each iteration:
  - Look for atom/column ϕ_j most closely resembling **all** the measurement vectors $\boldsymbol{y}_k$
  - Add this atom to estimated support
  - Remove the atom contribution to the measurements (-> approximation only)
    If $\mathcal{S}_t$ = estimated support at iteration t => build proxy for

$$\boldsymbol{\Phi}_{\mathcal{S}\setminus\mathcal{S}_t}(\boldsymbol{x}_k)_{\mathcal{S}\setminus\mathcal{S}_t} = \sum_{j \in \mathcal{S}\setminus\mathcal{S}_t} (\boldsymbol{x}_k)_j \boldsymbol{\phi}_j$$

Outline

- Introduction to compressive sensing
- Support recovery algorithms
- Multiple measurement vector signal models
- **Analysis of SOMP with noise**
- SOMP with noise stabilization
- Conclusion

Analysis of SOMP with noise
Objective & Main quantities

$$\boldsymbol{y}_k = \boldsymbol{\Phi}\boldsymbol{x}_k + \boldsymbol{e}_k \quad \text{where} \quad \boldsymbol{e}_k \sim \mathcal{N}(\boldsymbol{0}, \sigma_k^2 \boldsymbol{I}_{m \times m})$$

- **Noisy** signal model with **additive Gaussian** measurement noise

- General objective: understand how the additive Gaussian noise affect the performance of SOMP

- Main results:
  - Upper bound on the probability that SOMP fails (i.e., picks an incorrect atom) for *s+1* iterations
  - Corresponding minimal value of *K* for a prescribed maximum probability of failure
  - Numerical results confirm the theory is (mostly) correct

Analysis of SOMP with noise Quantities without noise

$$\boldsymbol{y}_k = \boldsymbol{\Phi}\boldsymbol{x}_k + \boldsymbol{e}_k \quad \text{where} \quad \boldsymbol{e}_k \sim \mathcal{N}(\boldsymbol{0}, \sigma_k^2 \boldsymbol{I}_{m\times m})$$

- Iteration *t*, quantities from the **noiseless** case:

$\gamma_c^{(t,\boldsymbol{P})}$ = Highest value of SOMP metric for **correct** atoms

$\gamma_i^{(t,\boldsymbol{P})}$ = Highest value of SOMP metric for **incorrect** atoms

$$\frac{\gamma_c^{(t,\boldsymbol{P})}}{\gamma_i^{(t,\boldsymbol{P})}} \geq \Gamma > 1 \text{ for any iteration } t.$$

$\Gamma > 1$ => correct decisions in noiseless case

$$\gamma_c^{(t,\boldsymbol{P})} \geq \psi \tau_X \text{ for any iteration } t$$

$$\Gamma = \frac{1 - \delta_{|\mathcal{S}|+1}}{\delta_{|\mathcal{S}|+1}\sqrt{|\mathcal{S}|}}$$

$$\tau_X = \min_{j \in \mathcal{S}} \sum_{k=1}^{K} |X_{j,k}|$$

$$\psi = \frac{(1 - \delta_{|\mathcal{S}|})(1 + \delta_{|\mathcal{S}|})}{1 + \sqrt{|\mathcal{S}|}\,\delta_{|\mathcal{S}|}}$$

$$\gamma_c^{(t,\boldsymbol{P})} - \gamma_i^{(t,\boldsymbol{P})} \geq \psi \tau_X \left(1 - \frac{1}{\Gamma}\right)$$

Upper bound prob. failure

$$\boldsymbol{y}_k = \boldsymbol{\Phi} \boldsymbol{x}_k + \boldsymbol{e}_k \quad \text{where} \quad \boldsymbol{e}_k \sim \mathcal{N}(\boldsymbol{0}, \sigma_k^2 \boldsymbol{I}_{m \times m})$$

$$\xi := \underbrace{\left(1 - \frac{1}{\Gamma}\right) \psi}_{=:\alpha} \mathrm{SNR}_{\min} - \underbrace{\sqrt{\frac{2}{\pi}}}_{=:\beta} \omega_\sigma \qquad \mathcal{C}_s := \sum_{t=0}^{s} \binom{|\mathcal{S}|}{t}$$

- **Upper bound on the probability of error** of SOMP for $|\mathcal{S}|$ iterations

$$n \mathcal{C}_{|\mathcal{S}|+1} \exp\left[-\frac{1}{8} K \xi^2\right]$$

- Main interpretations:
  - $\xi < 0$ => probability of failure might be 1 as $K \to \infty$
  - Both meas. matrix $\boldsymbol{\Phi}$ and SNR should be « good » enough when compared to noise
  - Prob. failure decreases exponentially with K if $\xi > 0$

- Detailed interpretation of each quantity on next slide

Min. value of K for given probability of error (1)

$$\boldsymbol{y}_k = \boldsymbol{\Phi}\boldsymbol{x}_k + \boldsymbol{e}_k \quad \text{where} \quad \boldsymbol{e}_k \sim \mathcal{N}(\boldsymbol{0}, \sigma_k^2 \boldsymbol{I}_{m\times m})$$

- **Upper bound on the probability of error** for $|\mathcal{S}|$ iterations : $n\mathcal{C}_s \exp\left[-\frac{1}{8}K\xi^2\right]$
- **Minimum value of** ***K*** to achieve **probability of error** p_{err} for $|\mathcal{S}|$ iterations

$$K_{\min}(p_{\mathrm{err}}) \le \frac{8}{\xi^2}\log\left(\frac{n\mathcal{C}_{|\mathcal{S}|-1}}{p_{\mathrm{err}}}\right) \quad \text{with } \xi = \alpha\ \mathrm{SNR_m} - \beta\omega_\sigma$$

- α : to what extent is $\boldsymbol{\Phi}$ appropriately designed ($0 < \alpha \le 1$)?
- $\mathrm{SNR_m}$: signal-to-noise ratio for all the *K* channels
- $\alpha\ \mathrm{SNR_m}$: term related to SNR and quality of meas. matrix $\boldsymbol{\Phi}$
- ω_σ : penalty depending on noise std. dev. uniformity (-> sparsity of $\boldsymbol{\sigma}$)
- β : theoretical constant ($\beta \le \sqrt{2/\pi}$)
- $\beta\omega(\sigma_1, \ldots, \sigma_K)$: noise-related penalty on robustness without noise
- $n\mathcal{C}_{|\mathcal{S}|-1}$: increases with # of atoms *n* and support size $|S|$
  - Theoretical expression is not sharp

Min. value of K for given probability of error (2)

$$\boldsymbol{y}_k = \boldsymbol{\Phi}\boldsymbol{x}_k + \boldsymbol{e}_k \quad \text{where} \quad \boldsymbol{e}_k \sim \mathcal{N}(\boldsymbol{0}, \sigma_k^2 \boldsymbol{I}_{m\times m})$$

- **Minimum value of** K to achieve **probability of error** $p_{\rm err}$:

$$K_{\min}(p_{\rm err}) := \frac{8}{\xi^2}\log\left(\frac{n\mathcal{C}_{|\mathcal{S}|-1}}{p_{\rm err}}\right) \qquad \xi = \alpha\,\mathrm{SNR}_{\rm m} - \beta\omega_\alpha$$

- Rewrites

$$\boxed{K_{\min}(p_{\rm err}) := \frac{8}{(\alpha\,\mathrm{SNR}_{\min} - \omega_\sigma\beta)^2}\left(\gamma - \log p_{\rm err}\right)}$$

with $\gamma := \log\left(n\mathcal{C}_{|\mathcal{S}|-1}\right)$

- Useful for simulations

Simulation framework

- **Goal:** Validate theoretical analysis
- **Method:** Carry out simulations and compare results with formula

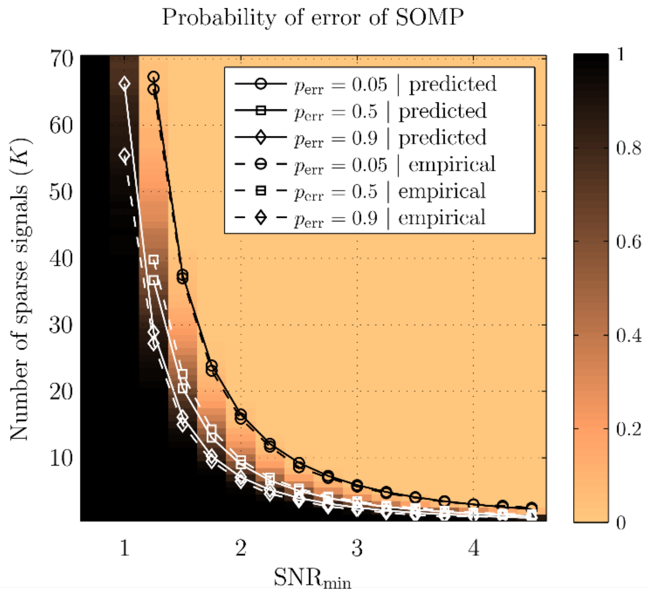
$$K_{\min}(p_{\text{err}}) := \frac{8}{(\alpha \, \text{SNR}_{\min} - \omega_\sigma \beta)^2} (\gamma - \log p_{\text{err}})$$

  - Identify the values of α , β, and γ on the basis of simulations.
  - Assess whether theoretical curve fits simulation curves
  - Assess whether identified values are coherent with theory

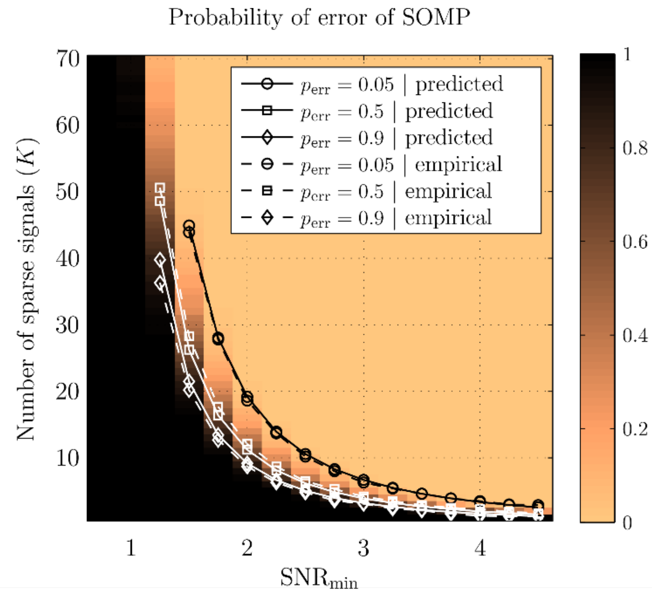
- Detailed signal model is not described here
- Identification procedure not discussed either

Simulations – Results (1)

$$K_{\min}(p_{\text{err}}) := \frac{8}{(\alpha\,\text{SNR}_{\min} - \omega_\sigma \beta)^2}(\gamma - \log p_{\text{err}})$$



(a) $|\mathcal{S}| = 10$ – Identified parameters: $\alpha = 1.0535$, $\beta = 0.54045$, and $\gamma = 2.0741$.



(b) $|\mathcal{S}| = 20$ – Identified parameters: $\alpha = 1.0535$, $\beta = 0.58682$, and $\gamma = 2.5451$.

Question 1: Do theoretical curves fit empirical ones?

☺ **YES** ☺

Simulations – Results (2)

$$K_{\min}(p_{\text{err}}) := \frac{8}{(\alpha\ \text{SNR}_{\min} - \omega_\sigma \beta)^2} (\gamma - \log p_{\text{err}})$$

Question 2: Are the identified values coherent with the theory?

$|\mathcal{S}| = 10$ – Identified parameters:
$\alpha = 1.0535,\ \beta = 0.54045,\ \text{and}\ \gamma = 2.0741$

$|\mathcal{S}| = 20$ – Identified parameters:
$\alpha = 1.0535,\ \beta = 0.58682,\ \text{and}\ \gamma = 2.5451$

- α should be <= 1 in theory but discrepancy is OK
- β is lower than $\sqrt{2/\pi} \simeq 0.7979$
- It can be shown that γ is way too low wrt the theory
  - but proof method explains why
  - and $\gamma := \log\left(n\mathcal{C}_{|\mathcal{S}|-1}\right)$ increases with support cardinality $|\mathcal{S}|$
  - See « Future work » in the thesis

Analysis mostly OK for α and β

The contributions so far

- Contribution:
  - Thorough analysis of noiseless and noisy SOMP (theory + simulations)
- Related publications:
  - *"On The Exact Recovery Condition of Simultaneous Orthogonal Matching Pursuit"*, IEEE Signal Processing Letters, vol. 23, no. 1, 2016
  - *"Improving the Correlation Lower Bound for Simultaneous Orthogonal Matching Pursuit"*, IEEE Signal Processing Letters, vol. 23, no. 11, 2016
  - *"On the Noise Robustness of Simultaneous Orthogonal Matching Pursuit"*, IEEE Transactions on Signal Processing, vol. 65, no. 4, 2017.

Outline

- Introduction to compressive sensing
- Support recovery algorithms
- Multiple measurement vector signal models
- Analysis of SOMP with noise
- **SOMP with noise stabilization**
- Conclusion

SOMP with noise stabilization (SOMP-NS)

$$\boldsymbol{y}_k = \boldsymbol{\Phi}\boldsymbol{x}_k + \boldsymbol{e}_k \quad \text{where } \boldsymbol{e}_k \sim \mathcal{N}(\boldsymbol{0}, \sigma_k^2 \boldsymbol{I})$$

Require: $\boldsymbol{Y} \in \mathbb{R}^{m \times K}$, $\boldsymbol{\Phi} \in \mathbb{R}^{m \times n}$, $s \geq 1$, $\{q_k\}_{1 \leq k \leq K}$

Initialization: $\boldsymbol{R}^{(0)} \leftarrow \boldsymbol{Y}$ and $\mathcal{S}_0 \leftarrow \emptyset$

$t \leftarrow 0$

while $t < s$ **do**

  Determine the column of $\boldsymbol{\Phi}$ to be included in the support:

  $j_t \leftarrow \text{argmax}_{1 \leq j \leq n} \sum_{k=1}^{K} |\langle \boldsymbol{\phi}_j, \boldsymbol{r}_k^{(t)} \rangle| q_k$

  Update the support : $\mathcal{S}_{t+1} \leftarrow \mathcal{S}_t \cup \{j_t\}$

  Projection of each measurement vector onto $\mathcal{R}(\boldsymbol{\Phi}_{\mathcal{S}_{t+1}})^{\perp}$:

  $\boldsymbol{R}^{(t+1)} \leftarrow (\boldsymbol{I} - \boldsymbol{\Phi}_{\mathcal{S}_{t+1}} \boldsymbol{\Phi}_{\mathcal{S}_{t+1}}^{+})\boldsymbol{Y}$

  $t \leftarrow t + 1$

end while

return $\mathcal{S}_s$ {Support at last step}

Idea: The SNRs of the measurement vectors are unequal ➡ weight the impact of each measurement vector according to its reliability.

$$\sum_{k=1}^{K} |\langle \boldsymbol{\phi}_j, \boldsymbol{r}_k^{(t)} \rangle| \quad \Rightarrow \quad \sum_{k=1}^{K} |\langle \boldsymbol{\phi}_j, \boldsymbol{r}_k^{(t)} \rangle| q_k$$

Question : What are the optimal weights q_k ?

(Our) Answer: Resort to the theory and find how to minimize an upper bound on the probability of SOMP-NS failing to perform correct decisions.

Theoretical probability of failure
Optimal weights

- If we assume $|X_{j,k}| \simeq c_k |X_{j,1}|$ $(c_k > 0)$ for $(j,k) \in [n] \times [K]$

$$\boxed{q_k = c_k / \sigma_k^2}$$

Formula stems from our analysis of SOMP with noise

Question 1: Does SOMP-NS yield improvements?

Question 2: Theoretically optimal weights = truly optimal weights?

➡ Simulations

Theoretically opt. Weights vs. truly optimal ones

- $K = 2, \;\; \boldsymbol{\sigma} := (\sigma_1, \sigma_2) = (\cos(\theta_\sigma), \sin(\theta_\sigma))$ and $\boldsymbol{q} := (q_1, q_2) = (\cos(\theta_q), \sin(\theta_q))$
- Grid $\mathcal{G} := \mathcal{G}_\sigma \times \mathcal{G}_q$ of values for θ_σ and θ_q $\rightarrow$ evaluate the probability of SOMP-NS succeeding in recovering the support $\mathcal{S}$ in exactly $|\mathcal{S}|$ iterations for each 2-tuple $(\theta_\sigma, \theta_q)$

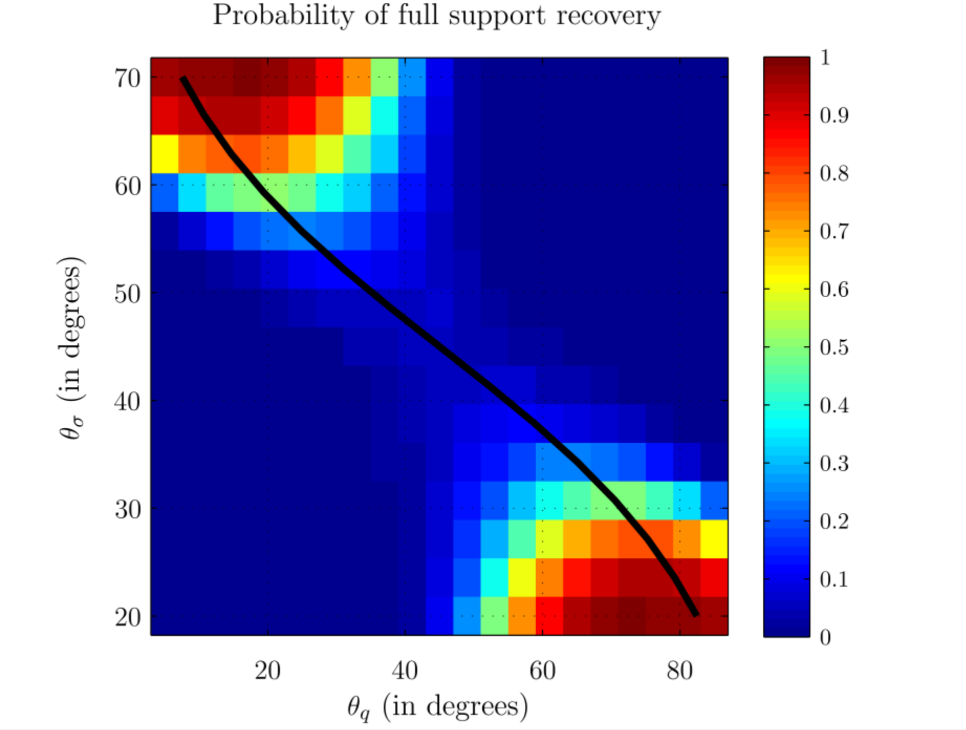


Figure 1: Simulation results for simulation setup 3.2 — The black line represents the analytically optimal weights given by $q_k = 1/\sigma_k^2$.

Theoretically opt. Weights vs. truly optimal ones

- $K = 2$, $\boldsymbol{\sigma} := (\sigma_1, \sigma_2) = (\cos(\theta_\sigma), \sin(\theta_\sigma))$ and $\boldsymbol{q} := (q_1, q_2) = (\cos(\theta_q), \sin(\theta_q))$
- Grid $\mathcal{G} := \mathcal{G}_\sigma \times \mathcal{G}_q$ of values for θ_σ and θ_q $\rightarrow$ evaluate the probability of SOMP-NS succeeding in recovering the support $\mathcal{S}$ in exactly $|\mathcal{S}|$ iterations for each 2-tuple $(\theta_\sigma, \theta_q)$

- n = 1000, real and complex signal models (random sign or random phase), $\|\boldsymbol{\sigma}\|_2^2 = 1$

- Two different signal patterns

Signal patterns

SP1: signal pattern with non-zero entries of **X** with equal moduli

$$\boldsymbol{X} = \begin{bmatrix} 1 & -1 & 1 \\ 0 & 0 & 0 \\ \vdots & \vdots & \vdots \\ 0 & 0 & 0 \\ -1 & -1 & -1 \\ 1 & 1 & -1 \\ \vdots & \vdots & \vdots \end{bmatrix}$$

SP2: signal pattern with Gaussian non-zero entries (N(0,1)), common for all columns, that are then normalized

$$\boldsymbol{X} = \begin{bmatrix} 3.36 & -3.36 & 3.36 \\ 0 & 0 & 0 \\ \vdots & \vdots & \vdots \\ 0 & 0 & 0 \\ -1.42 & -1.42 & -1.42 \\ 1 & 1 & -1 \\ \vdots & \vdots & \vdots \end{bmatrix}$$

Similar non-zero moduli

$$q_k = 1/\sigma_k^2$$

Dissimilar non-zero moduli

Theoretically opt. Weights vs. truly optimal ones (Results)

$q_k = 1/\sigma_k^2$

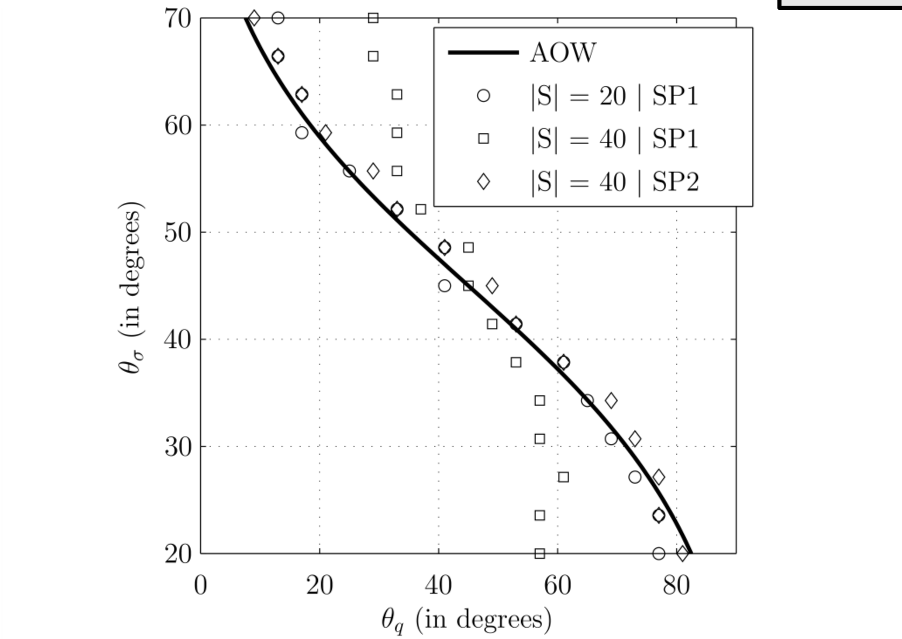


Figure 1: Optimal weighting angles — Simulation setups 1.1 to 1.3 — The black continuous line represents the analytically optimal weights (AOW) given by $q_k = 1/\sigma_k^2$.

$\mathbb{R}$

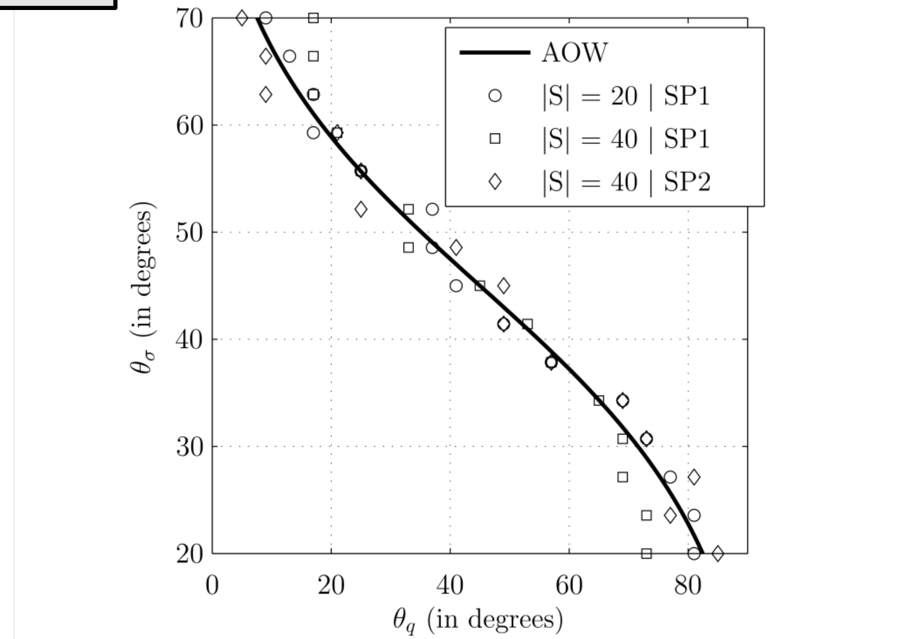


Figure 1: Optimal weighting angles – Simulation setups 2.1 to 2.3 – The black, continuous line represents the analytically optimal weights (AOW) given by $q_k = 1/\sigma_k^2$.

$\mathbb{C}$

Outline

- Introduction to compressive sensing
- Support recovery algorithms
- Multiple measurement vector signal models
- Analysis of SOMP with noise
- SOMP with noise stabilization
- **Conclusion**

Conclusion

- Contributions:
  - Analysis of SOMP with and without noise
  - Proposal and analysis of SOMP-NS
  - Numerical validation for both contributions

Thank you for your attention!

Outline

- Introduction to compressive sensing
- Support recovery algorithms
- Multiple measurement vector signal models
- Analysis of SOMP with noise
- SOMP with noise stabilization
- Conclusion

- **Backup slides**

Orthogonal matching pursuit

$$\boldsymbol{y} = \boldsymbol{\Phi x} = \sum_{j \in \mathcal{S}} x_j \boldsymbol{\phi}_j$$

Require: $\boldsymbol{y} \in \mathbb{R}^m$, $\boldsymbol{\Phi} \in \mathbb{R}^{m \times n}$, $s \geq 1$

1: Initialization: $\boldsymbol{r}^{(0)} \leftarrow \boldsymbol{y}$ and $\mathcal{S}_0 \leftarrow \emptyset$

2: $t \leftarrow 0$

3: **while** $t < s$ **do**

4: Determine the atom of $\boldsymbol{\Phi}$ to be included in the support:
$j_t \leftarrow \mathrm{argmax}_{j \in [n]} \, |\langle \boldsymbol{r}^{(t)}, \boldsymbol{\phi}_j \rangle|$

5: Update the support : $\mathcal{S}_{t+1} \leftarrow \mathcal{S}_t \cup \{j_t\}$

6: Projection of the measurement vector onto $\mathcal{R}(\boldsymbol{\Phi}_{\mathcal{S}_{t+1}})$:
$\boldsymbol{y}^{(t+1)} \leftarrow \boldsymbol{\Phi}_{\mathcal{S}_{t+1}} \boldsymbol{\Phi}^{+}_{\mathcal{S}_{t+1}} \boldsymbol{y}$

7: Projection of the measurement vector onto $\mathcal{R}(\boldsymbol{\Phi}_{\mathcal{S}_{t+1}})^{\perp}$:
$\boldsymbol{r}^{(t+1)} \leftarrow \boldsymbol{y} - \boldsymbol{y}^{(t+1)}$

8: $t \leftarrow t + 1$

9: **end while**

10: **return** $\mathcal{S}_s$ {Support at last step}

Simulations – Results

$$\boldsymbol{\sigma} = (\sigma_{\text{odd}}, \sigma_{\text{even}}, \sigma_{\text{odd}}, \sigma_{\text{even}}, \dots) \text{ where } r_\sigma = \sigma_{\text{even}}/\sigma_{\text{odd}}$$

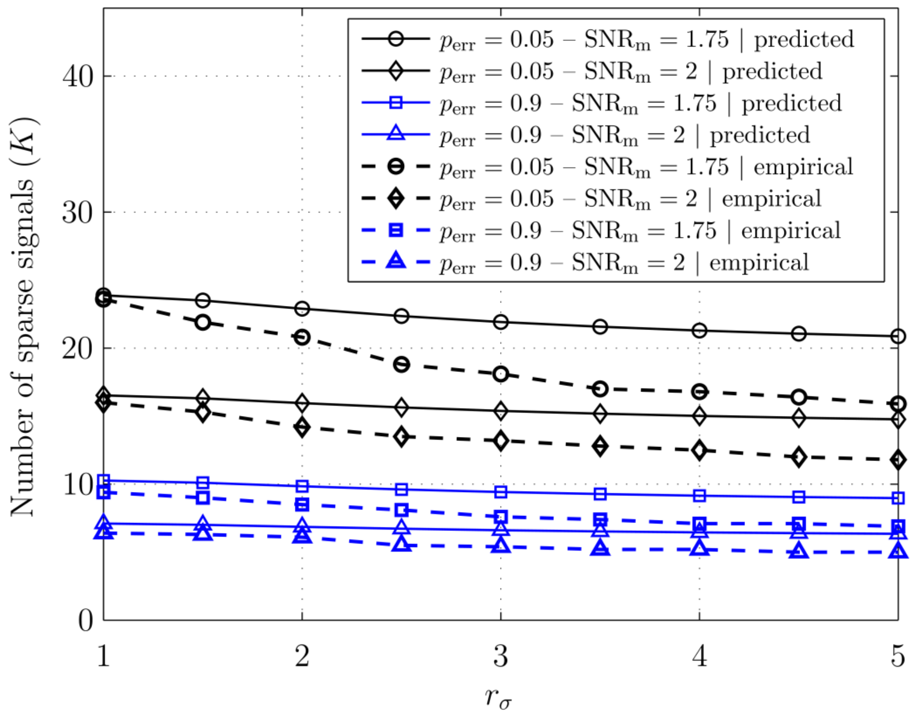


Figure 1: Levels sets of the probability of SOMP comitting at least one error when performing the joint full support recovery — $|\mathcal{S}| = 10$

$$\omega_\sigma = \frac{1}{\sqrt{2}}(r_\sigma + 1)/\sqrt{r_\sigma^2 + 1}$$