Under-determined Blind Source Localization by Exploiting Microphone Array Geometry.

M. Umair Khan
2017-MS-CE-15

Department of Computer Science and Engineering
University of Engineering and Technology, Lahore

umair.khan.uet59@gmail.com

Thesis Supervisor
Dr. Tania Habib

October 23, 2019

Outline

Cocktail Party Problem

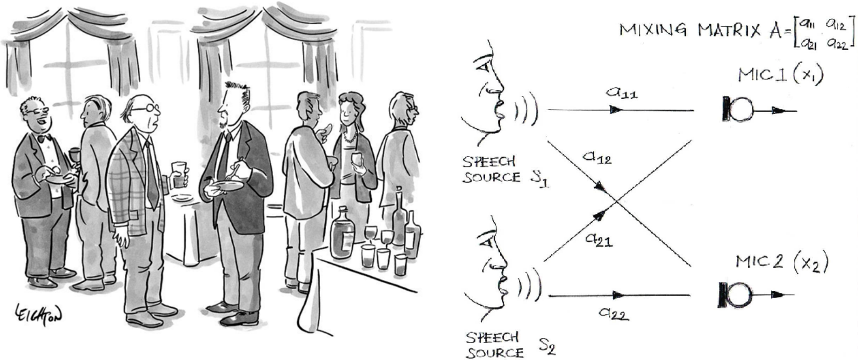


Figure: Cocktail party and the mixing system [1, 2]

Blind Source Separation solves the cocktail party problem.

Independent Component Analysis is a method used for blind source separation.

Introduction to Blind Source Separation

The problem is modelled as:

$$\boldsymbol{x} = \boldsymbol{A}\boldsymbol{s}$$

$$\hat{\boldsymbol{s}} = \boldsymbol{W}\boldsymbol{x}$$

where,

$$\boldsymbol{W} = \boldsymbol{A}^{-1}$$

$\boldsymbol{x}$ = observed source signals.
$\boldsymbol{A}$ = mixing matrix.
$\boldsymbol{W}$ = unmixing matrix.
$\boldsymbol{s}$ = original source signals.
$\hat{\boldsymbol{s}}$ = estimated source signals.

for a 2-input 2-output system:

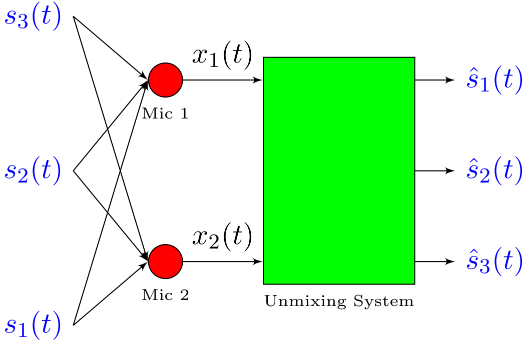
$$\boldsymbol{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$

Assumptions:

1. $\boldsymbol{A}$ is invertible.
2. $\boldsymbol{s}$ is statistically independent.
3. $\boldsymbol{s}$ is non-gaussian.

Finding *Direction of Arrival (DoA)* is central to solving the BSS problem.

Under-determined Blind Source Separation



If the acoustic sources out numbers the number of microphones, the system is called *Under-determined Blind Source Separation.*

Blind Source Localization

- Used for solving *permutation ambiguity* in BSS.
- Calculated using Time Difference of Arrival (TDOA) in **far field**.

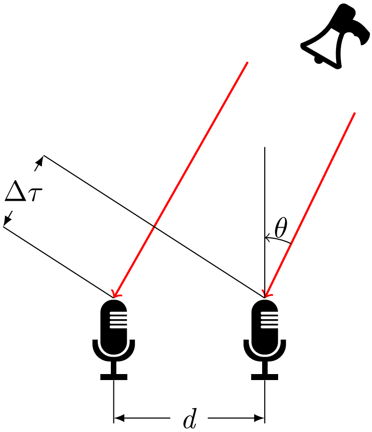
$$\boxed{\theta = sin^{-1}\left(\frac{\Delta\tau \times v}{d}\right)}$$

where,

$\Delta\tau$ = TDOA

v = speed of sound waves

d = mic separation



Under-determined Blind Source Localization is localization of acoustic sources greater than the number of available microphones.

Related Work

Paper ref.	**Sensor type**	**Reverb. Env.**	**Field of view**	**Separation**	**BSS as baseline**	**Mic. separation**
Nogueria et al. 2015 [3]	Distrubuted	No	Far	Determined	Yes	unknown
Wang et al. 2016 [4]	Distrubuted	No	Far + Near	Determined	No	variable
Brendel et al. 2017 [5]	Distrubuted	Yes	Far	Under-Determined	No	20 cm
Brendel et al. 2018 [6]	Distrubuted	Yes	Far	Determined	No	20 cm
Proposed Work	**Condensed**	**Yes**	**Far**	**Under-Determined**	**Yes**	**9.26 cm**

Research Gap

- BSS algorithms are generally limited to two speakers only and are geometrically restrictive.

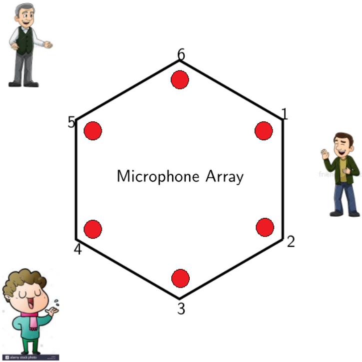
Problem Statement

To design and develop an algorithm to solve the problem of *Under-determined Blind Source Localization* using BSS algorithm as the baseline by exploiting geometry of the microphone array. The existing BSS algorithms are geometrically restrictive.

Scope

1. Support for variable geometries (hexagonal, triangular, circular, square etc.)
2. Recordings in various surroundings (meeting room, reverberant environment, studio environment).
3. Localization of more than two speakers (under-determined case).
4. Continuously moving speakers.
5. Speakers periodically changing their positions.

Experimental Setup



Sample

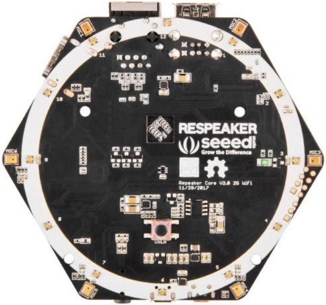
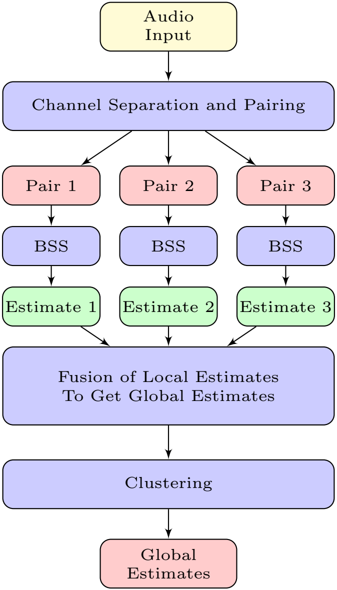


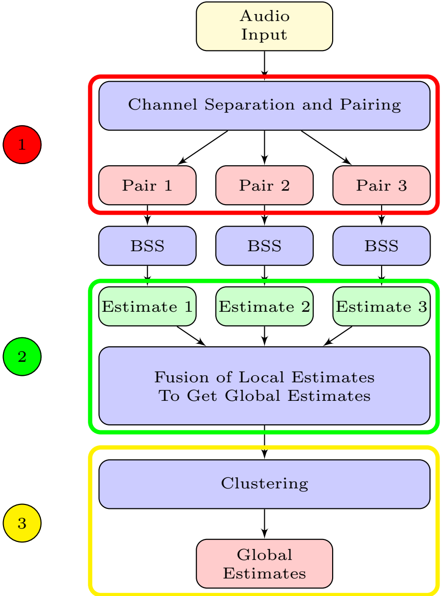
Figure: ReSpeaker Core v2.0 from Seeed Studio [7].

1. Six microphone array.
2. Quad-Core Cortex-A7 up to 1.5 GHz.
3. 1 GB RAM.
4. Runs Debian® or Android®.

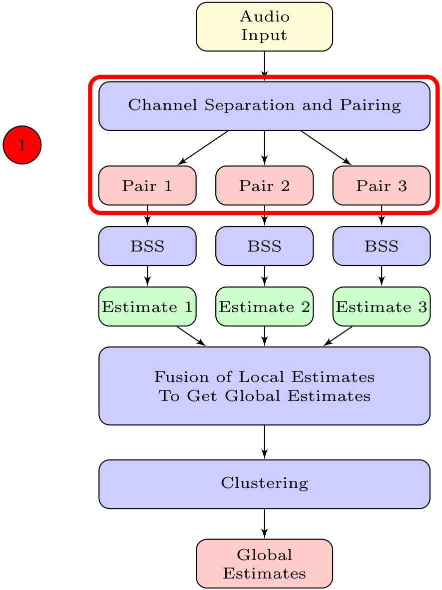
Methodology



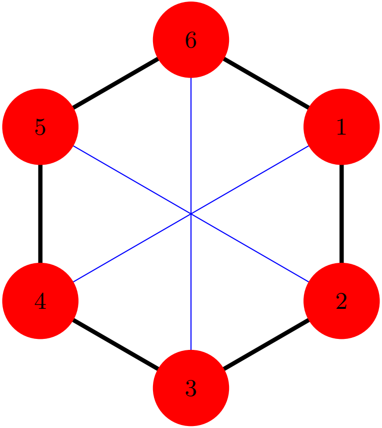
Methodology



Methodology

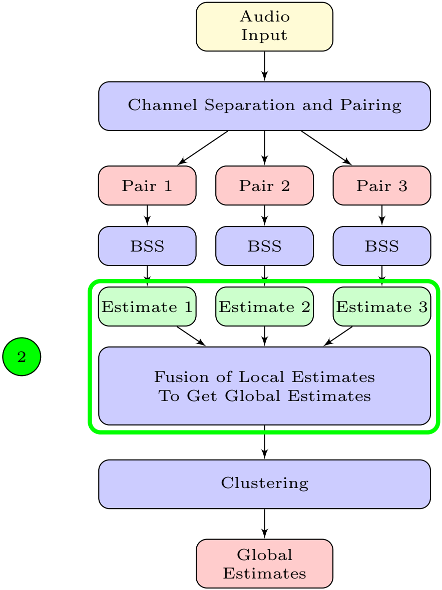


Channel separation and pairing



- [6-3] = Pair 1
- [5-2] = Pair 2
- [4-1] = Pair 3

Methodology



Front-Back Disambiguation Problem

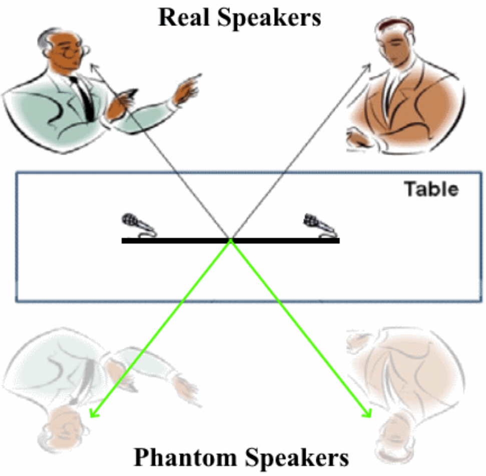
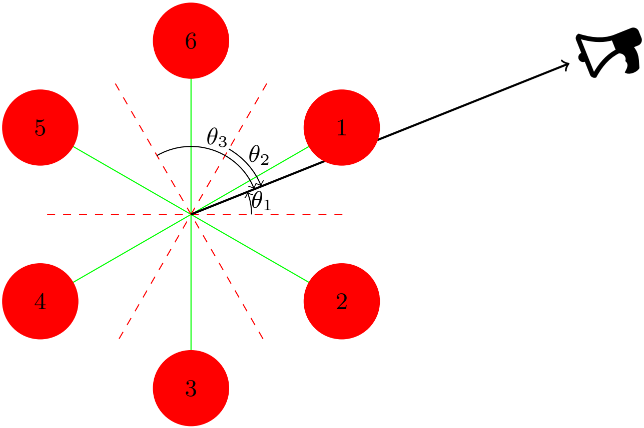


Figure: A 2-speaker 2-microphone setup gives four location estimates. Picture credits [8].

Finding global estimates from local estimates



- $\theta_{1\_global} = \theta_1 + \theta_{1\_axis}$
- $\theta_{2\_global} = \theta_2 + \theta_{2\_axis}$
- $\theta_{3\_global} = \theta_3 + \theta_{3\_axis}$

Global Coherence Map

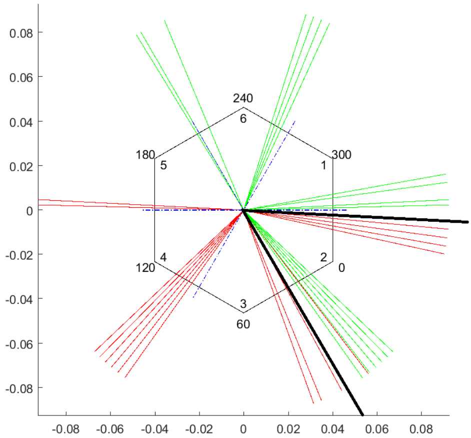
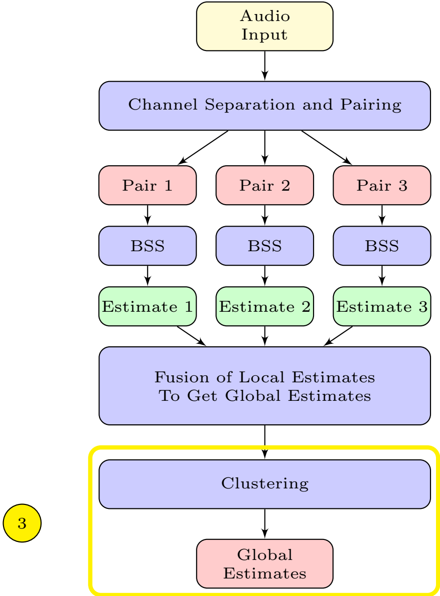


Figure: Local estimates plotted after rotation to get GCM.

Methodology



Comparison of Clustering Algorithms

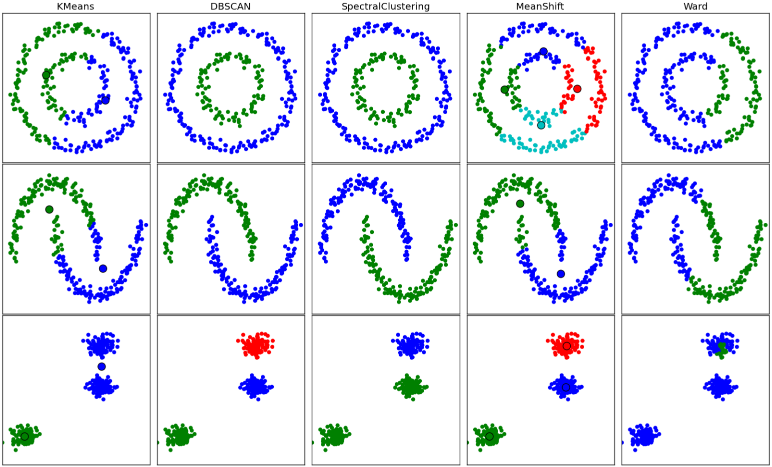


Figure: DBSCAN out performs all other algorithms [9]

Comparison of Clustering Algorithms

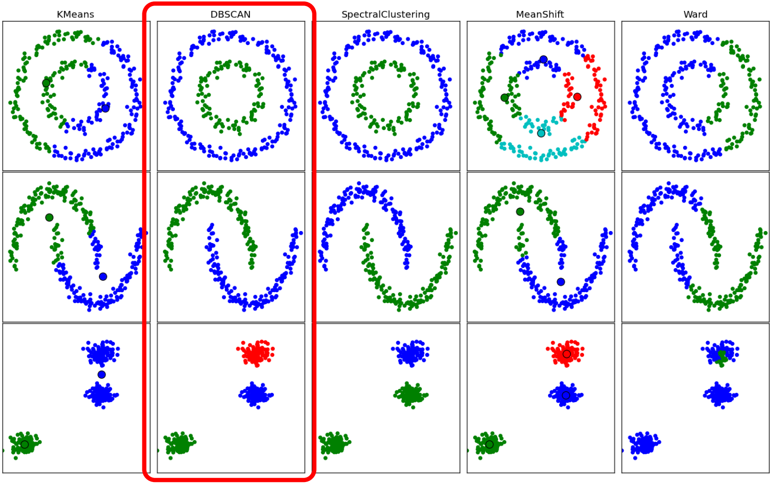


Figure: DBSCAN out performs all other algorithms [9]

DBSCAN clustering

- Unsupervised learning algorithm.
- Doesn't need to know the number of clusters before-hand.
- Takes two parameters as input:
  - `eps` - Maximum allowable distance between two points.
  - `min_points` - Minimum number of points that form a cluster.

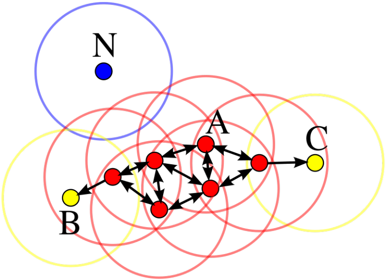
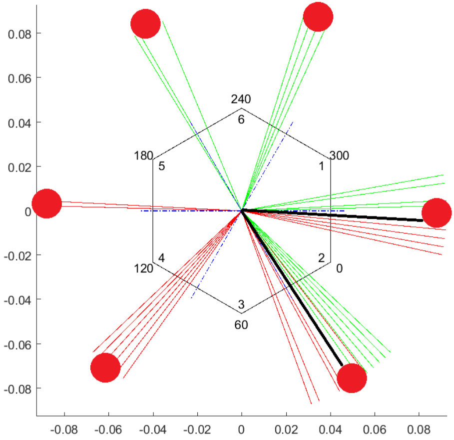


Figure: DBSCAN clustering [10]

Clustering Results



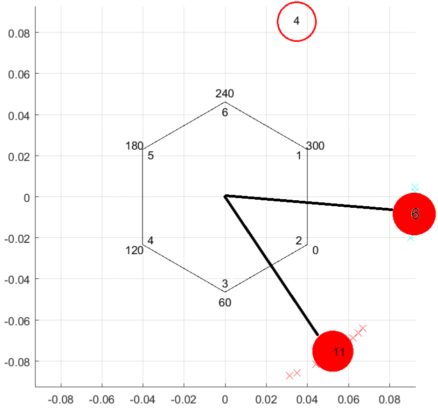
Challenges:

1. Two or more clusters of same size.
2. Phantom sources gather to form a fake cluster.

Proposed Solutions:

1. Pick the cluster that is most concentrated.
2. As a cluster is detected, remove its phantoms.

Improved GCM



Mathematical Representation

Observation vector

$$\boldsymbol{\theta} = \{\theta_1, \theta_2, \theta_3, ..., \theta_n\}$$

Clustering algorithm clusters the observation vector into m clusters

$$\boldsymbol{\Theta}' = \{\boldsymbol{\Theta}'_1, \boldsymbol{\Theta}'_2, \boldsymbol{\Theta}'_3, ..., \boldsymbol{\Theta}'_m\}$$

where,

$$\boldsymbol{\Theta}'_k = \{\theta_{1k}, \theta_{2k}, ..., \theta_{jk}\}$$

$$\mu_k = \overline{\boldsymbol{\Theta}'_k} = \frac{1}{n_k} \sum_{i=1}^{n} \theta_{ik}$$

where n_k is the number of elements in the k^{th} cluster.

Mathematical Representation (contd.)

Next step is to sort the list of clusters to m clusters represented as:

$$\mathbf{\Theta}_{sorted} = \{\mathbf{\Theta}_1, \mathbf{\Theta}_2, \mathbf{\Theta}_3, \mathbf{\Theta}_m\}$$

Pick the first N clusters from the sorted cluster set, where, N is the number of active speakers:

$$\mathbf{\Theta}_N = \{\mathbf{\Theta}_{sorted} \mid 1 \leq sorted \leq N\}$$

In case, there are multiple clusters of same size, pick those who are more concentrated towards their mean.

$$\mathbf{\Theta}_{Nn} = \operatorname*{argmin}_{\mathbf{\Theta}_k} \sum_{\theta_{ik} \in \mathbf{\Theta}_k} \mid \theta_{ik} - \mu_k \mid$$

Mathematical Representation (contd.)

Finally, the location estimates for N active speakers are found as follows:

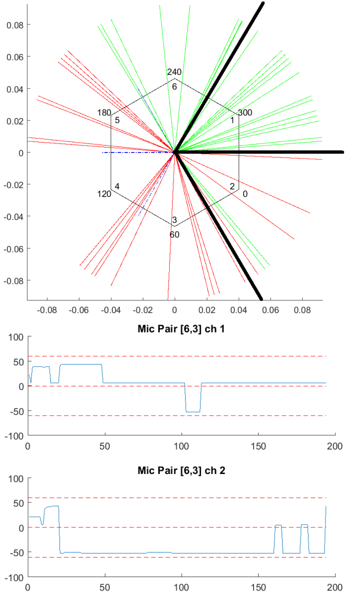
$$\boxed{\mu_N = \overline{\boldsymbol{\Theta}_N} = \frac{1}{n_N} \sum_{i=1,\theta_{iN} \in \boldsymbol{\Theta}_N}^{n} \theta_{iN}}$$

Results - Two speakers

Real location [deg.]	Estimated location [deg.]	Difference [deg.]	RMSE
30 ± 10	28.464	1.536	2.0845
90 ± 10	87.367	2.633	
30 ± 10	25.038	4.962	5.067
330 ± 10	335.19	5.19	
90 ± 10	90.21	0.21	3.68
150 ± 10	142.85	7.15	
100 ± 10	107.619	7.619	8.235
150 ± 10	141.148	8.852	
210 ± 10	206.106	3.894	3.939
270 ± 10	273.984	3.984	

Localization of Three Concurrent Speakers

1. Under-determined BSS.
2. Hexagonal is not the best geometry for this case.
   - For some microphone pairs, two sources align at real and phantom positions.
3. All three sources are captured only during a small window throughout the recording period.
   - Lack of state-of-the-art recording setup.
   - Noisy surroundings.
4. More occurring values gain weightage and suppress the less occurring values.



Solutions to the challenges on previous slide

1. Small window size isn't optimal for this geometry.
2. Take good length of the audio recording before making the final estimation (upto 10 sec).
3. Give equal weightage to every occurring estimated value.

Results - Three speakers

Real location [deg.]	Estimated location [deg.]	Difference [deg.]	RMSE
30 ± 10	25.84	4.16	11.8767
270 ± 10	290.99	20.99	
330 ± 10	319.52	10.48	
30 ± 10	24.69	5.31	11.4703
90 ± 10	99.933	9.933	
330 ± 10	349.168	19.168	
90 ± 10	271.931	181.931	96.207
210 ± 10	214.534	4.534	
330 ± 10	324.051	5.949	
30 ± 10	37.235	7.235	5.364
195 ± 10	191.546	3.454	
255 ± 10	260.405	5.405	
150 ± 10	144.487	5.513	8.37067
270 ± 10	264.847	5.153	
350 ± 10	335.554	14.446	

Tracking Moving Speakers

- Local estimates are divided into small overlapping windows (5 frames).
- Clustering algorithm is applied to each window separately and location is estimated.
- Estimates are plotted over time to get the tracks.

Tracking Two Moving Speakers

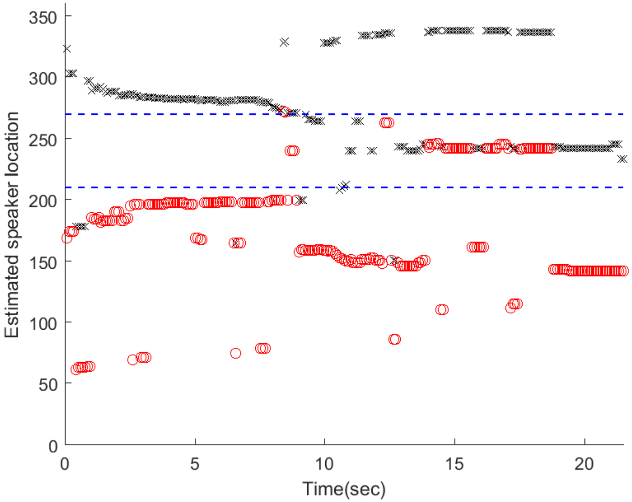


Figure: Estimated location plotted over time.

Tracking Two Moving Speakers

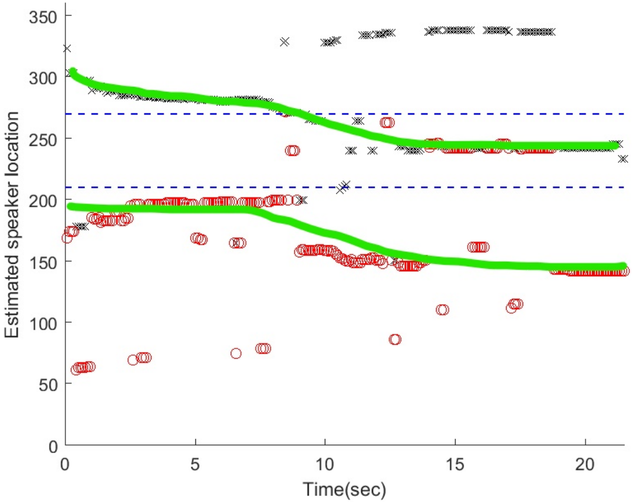


Figure: Estimated location plotted over time.

Summary

This research work:

1. Extends the BSS algorithm to hexagonal geometry.
2. Presents a solution to the well-know problem of Under-determined BSS.
3. Upto three simultaneously active speakers have been successfully localized.
4. The results can be fed to a tracker algorithm for speaker tracking.
5. The algorithm is self-adjusting to change in geometry.

References I

[1] Fine art america
https://fineartamerica.com/featured/two-men-make-introductions-at-a-party-robert-leighton.html

[2] BSS Mixing system
https://www.vocal.com/blind-signal-separation/blind-source-separation-for-noise-reduction-in-mobile/

[3] Nogueira, Luiz CF and Petraglia, Mariane R
Robust localization of multiple sound sources based on BSS algorithms
2015 IEEE 24th International Symposium on Industrial Electronics (ISIE)
pages 579-583

[4] Wang, Lin and Reiss, Joshua D and Cavallaro, Andrea
Over-determined source separation and localization using distributed microphones
2016 IEEE/ACM Transactions on Audio, Speech, and Language Processing
pages 1573-1588 vol.24 no.9

References II

[5] Brendel, Andreas and Kellermann, Walter
Localization of Multiple Simultaneously Active Sources in Acoustic Sensor Networks Using ADP.
2017 IEEE 7th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP) pages 1-5

[6] Brendel, Andreas, and Gannot, Sharon and Kellermann, Walter
Localization of Multiple Simultaneously Active Speakers in an Acoustic Sensor Network.
2018 IEEE 10th Sensor Array and Multichannel Signal Processing Workshop (SAM) pages 450-454

[7] ReSpeaker Core v2.0
`https://www.seeedstudio.com/ReSpeaker-Core-v2-0.html`

References III

[8] Dam, Hai Quang Hong and Ho, Hai and Le Ngo, Minh Hoang (2016)
Blind Speech Separation Using SRP-PHAT Localization and Optimal Beamformer in Two-Speaker Environments.
World Academy of Science, Engineering and Technology, International Journal of Computer, Electrical, Automation, Control and Information Engineering vol. 10, 1529–1533.

[9] Comparison of clustering algorithms
`https://medium.com/@jegasingamjeyanthasingam/comparing-clustering-algorithms-b55be9583619`

[10] DBSCAN clustering
`https://en.wikipedia.org/wiki/DBSCAN`

Thank You

Triangular Geometry

Didn't show good results because the microphone pairs weren't co-centric.

Actual location	Estimated location
30°	58.806°
270°	177.335°
330°	315.144°

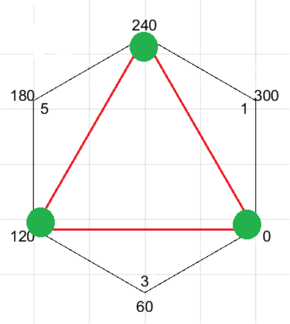
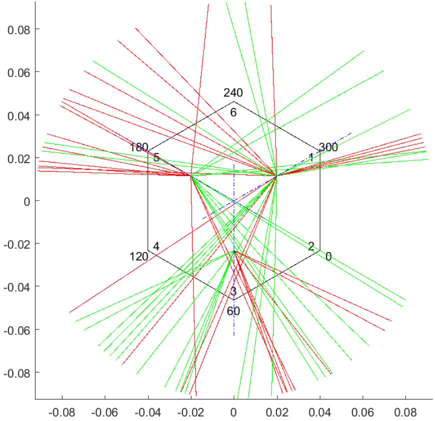
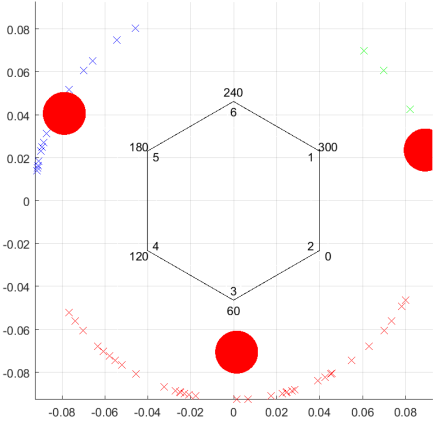


Figure: Making a triangular geometry from within hexagonal.

Triangular Geometry (contd.)





Tracking Three Speakers

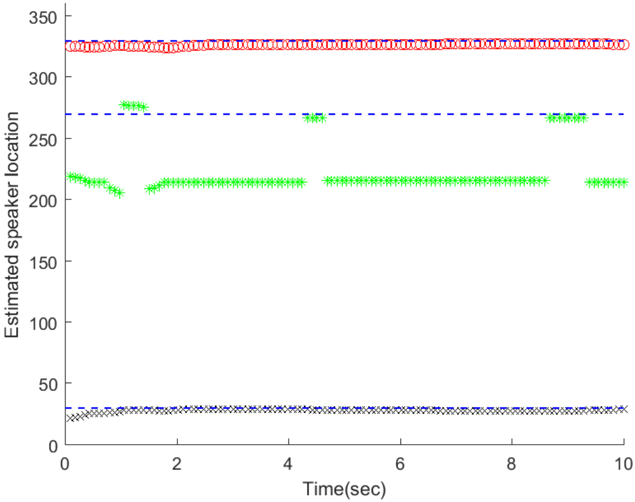
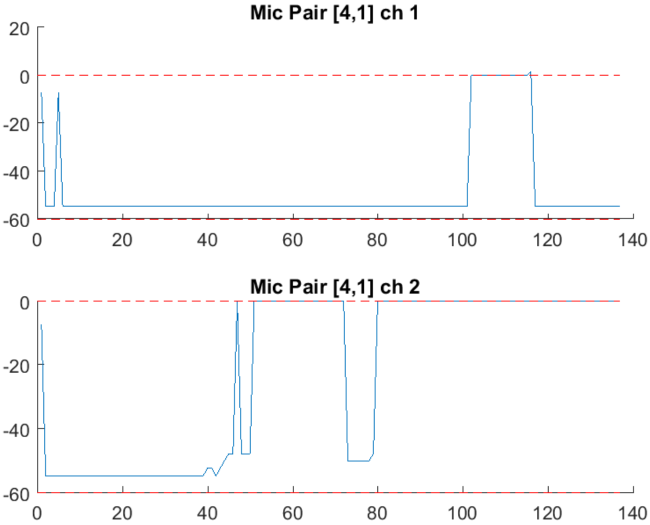


Figure: Estimated location plotted over time.

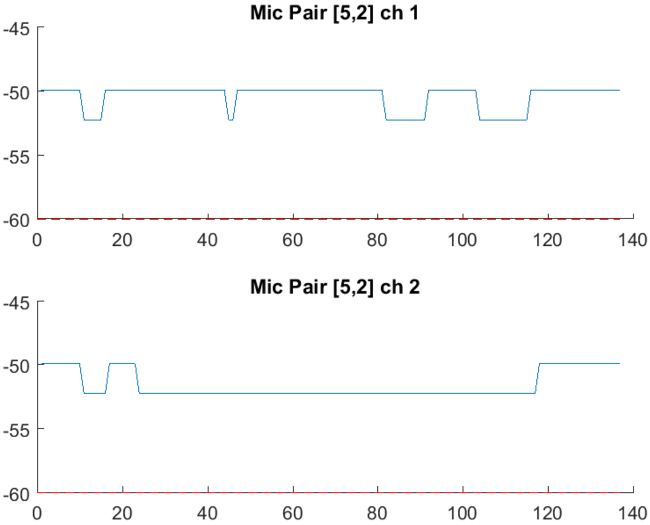
Future Work

- Making algorithm more robust.
- Capability to run online.
- Speaker Tracking by applying a tracking algorithm.
- Extension to three moving speakers.

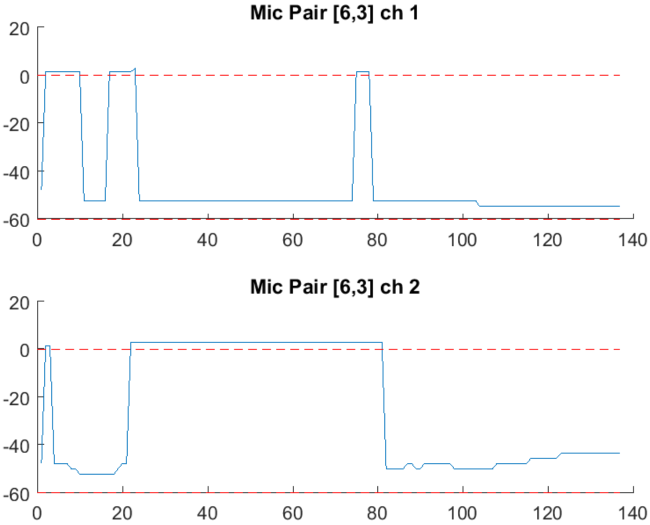
Local Estimates



Local Estimates (contd.)



Local Estimates (contd.)

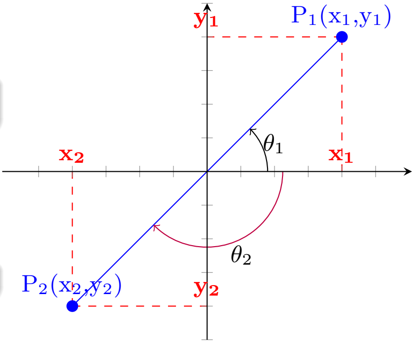


Global Estimates - Finding reference axis

$$\theta_1 = \tan^{-1} \frac{y_1}{x_1}$$

1. Can't work for all four quadrants.
2. `atan2` to be used.

```
>> atan2(y2, x2)
```



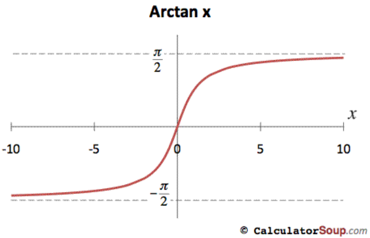


Figure: Graph of arctan [?]

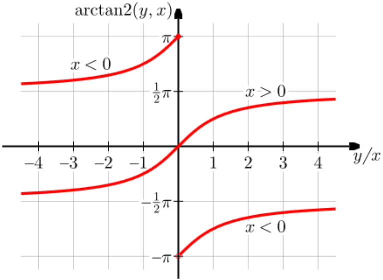
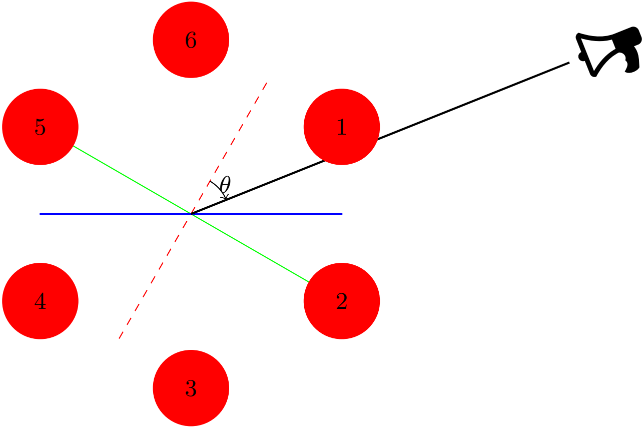


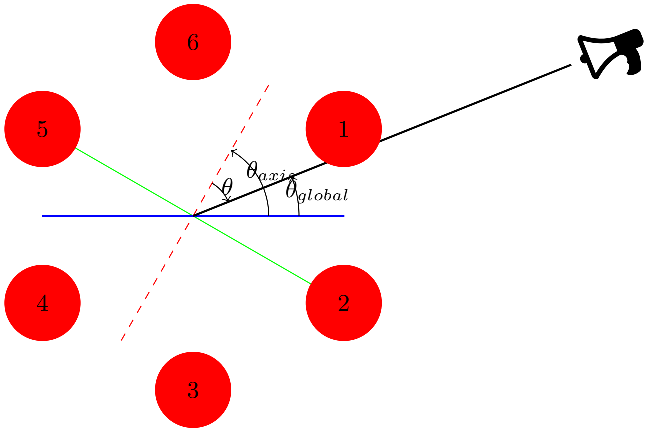
Figure: Graph of arctan2 [?]

Local Estimate



$$\theta_{global} = \theta + \theta_{axis}$$

Local Estimate



$$\theta_{global} = \theta + \theta_{axis}$$

Introduction to BSS

The problem is models as follows:

$$\boldsymbol{x} = \boldsymbol{A}\boldsymbol{s}$$

$\boldsymbol{x}$ = observed source signals.
$\boldsymbol{A}$ = mixing matrix.
$\boldsymbol{s}$ = original source signals.

for a 2-input 2-output system:

$$\boldsymbol{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$

1. $\boldsymbol{A}$ is invertible.
2. $\boldsymbol{s}$ is statistically independent.
3. $\boldsymbol{s}$ is non-gaussian.

$$\hat{\boldsymbol{s}} = \boldsymbol{W}\boldsymbol{x}$$

Using SVD,

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$$

$$\boldsymbol{W} = \boldsymbol{A}^{-1} = \boldsymbol{V}\boldsymbol{\Sigma}^{-1}\boldsymbol{U}^T$$

Covariance of $\boldsymbol{x}$ is given by,

$$\langle \boldsymbol{x}\boldsymbol{x}^T \rangle = \langle (\boldsymbol{A}\boldsymbol{s})(\boldsymbol{A}\boldsymbol{s})^T \rangle$$

$$= \langle (\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T)((\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T)^T \rangle$$

$$= \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T \langle \boldsymbol{s}\boldsymbol{s}^T \rangle \boldsymbol{V}\boldsymbol{\Sigma}\boldsymbol{U}^T$$

$$= \boldsymbol{U}\boldsymbol{\Sigma}^2\boldsymbol{U}^T$$

Whitening,

$$\boldsymbol{x} = \boldsymbol{A}\boldsymbol{s}$$

$$\boldsymbol{W} = \boldsymbol{A}^{-1} = \boldsymbol{V}\boldsymbol{\Sigma}^{-1}\boldsymbol{U}^{T}$$

so,

$$\hat{\boldsymbol{s}} = \boldsymbol{W}\boldsymbol{x}$$

We defind $\boldsymbol{x}_w$ as,

$$\boldsymbol{x}_w = (\boldsymbol{\Sigma}^{-1}\boldsymbol{U}^{T})\boldsymbol{x}$$

$$\hat{\boldsymbol{s}} = \boldsymbol{V}\boldsymbol{x}_w$$

We can estimate $\boldsymbol{V}$ using MLE as,

$$\boldsymbol{V} = \underset{\boldsymbol{V}}{\operatorname{argmin}} \sum_i \boldsymbol{H}[(\boldsymbol{V}\boldsymbol{x}\boldsymbol{w})_i]$$

where H is the entropy.

Now, it has reduced to an optimization problem. We can recover $\hat{\boldsymbol{s}}$ as follows:

$$\boldsymbol{x} = \boldsymbol{A}\boldsymbol{s}$$

$$\hat{\boldsymbol{s}} = \boldsymbol{W}\boldsymbol{x}$$

$$\boldsymbol{W} = \boldsymbol{A}^{-1} = \boldsymbol{V}\boldsymbol{\Sigma}^{-1}\boldsymbol{U}^{T}$$

Source signals can be estimated using observed data as:

$$\hat{\boldsymbol{s}} = \boldsymbol{W}\boldsymbol{x}$$